



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

교육학 박사학위 논문

소프트웨어 개발자의 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계

The Relationships among the Development Level of Expertise, Training and
Development, Qualification and Work Experience for Software Developers

2016년 2월

서울대학교 대학원

농산업교육과

곽 미 선

국 문 초 록

소프트웨어 개발자의 전문성 발달 수준과 교육훈련, 자격 및 직무경력과 관계

교육학 박사학위 논문

서울대학교 대학원, 2016년

곽 미 선

이 연구의 목적은 소프트웨어 개발자의 전문성 발달수준과 교육훈련, 자격 및 직무경력의 관계를 구명하는데 있다. 이 같은 목적을 위해 다음과 같은 연구목적을 설정하였다. 첫째, 소프트웨어 개발자의 전문성의 발달 수준을 구명한다. 둘째, 소프트웨어 개발자의 전문성 발달 수준에 따라 등급을 구분한 후, 등급별 특성을 구명한다. 셋째, 소프트웨어 개발자의 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계를 구명한다. 넷째, 소프트웨어 개발자의 전문성 등급 집단을 구분 짓는 교육훈련, 자격 및 직무경력의 판별력을 구명한다.

이 연구의 모집단인 소프트웨어 개발자는 소프트웨어 개발 생명주기(Life Cycle) 과정인 분석 및 설계, 구현 및 테스트와 전체 프로젝트를 총괄하는 관리 업무를 수행하는 사람이다. 한국소프트웨어산업협회(KOSA)에 등록된 기업 중 2014년 ‘소프트웨어 기술자 역량인정 체계 구축을 위한 연구’에서 활용한 250개 기업 리스트를 활용하여 표집 대상을 선정한 후 이 중 유의표집을 통해 20개의 기업을 선정하였다.

자료수집을 위한 소프트웨어 개발자의 전문성 발달 수준 측정도구는 선행연구와 국가직무능력표준, TOCIT 등에서 사용한 개발생명주기(life cycle)인 분석 및 설계, 구현, 테스트, 프로젝트 관리 영역을 바탕으로 연구자가 개발하였다. 예비조사를 통한 문항의 내적합치도 및 구인타당도 분석을 거쳐 본 조사에 사용할 문항을 재추출하였으며, 최종적으로 선정된 문항의 확인적 요인분석 결과를 통해 타당도와 신뢰도를 확보하였다.

자료수집은 2015년 11월 2일부터 11월 13일까지 2주간 진행하여 단순 직무를 수행하는 기능직을 제외한 526부가 최종 분석에 활용되었다. 수집된 자료는 각각의 연구목표에 따라 확인적 요인분석, 주성분분석, K-mean 군집분석, F검증, 다중회귀분석, 판별분석을 활용하여 분석하였다. 추리통계에 대한 통계적 유의성은 0.05를 기준으로 판단하였다.

또한 소프트웨어 개발자의 전문성 발달 수준에 대한 교육훈련, 자격 및 직무경력의 결정력을 분석하기 위해 학력과 전공으로 계산한 학교교육, 입직 후 상급학교 진학과 사내외 교육훈련으로 계산한 평생교육, 소지한 자격증의 수준, 유형, 개수로 계산한 자격증 소지 자격, 직급별 참여한 프로젝트 수행 개수로 계산한 프로젝트 수행 자격 및 직급별 종사년수와 소프트웨어 관련 여부로 계산한 직무경력으로 분석하였다. 또한, 노임단가 기준으로 활용되는 최종학력, 보유 자격증의 최고 수준, 종사년수 변인을 활용한 연구결과와 비교분석하였다.

이 연구의 결과는 다음과 같다.

첫째, 소프트웨어 개발자의 전문성 발달 수준은 분석 및 설계, 구현, 테스트 및 프로젝트 관리로 구성되었으며 평균 3.00이었다. 전문성 발달 수준이 높아짐에 따라 분석 및 설계, 프로젝트 관리의 중요도가 높아진다는 선행연구 결과를 반영하기 위해 주성분 분석을 실시하여 분석 및 설계(0.303), 구현(0.224), 테스트(0.223) 및 프로젝트 관리(0.250)로 영역별 가중치를 도출한 후 전문성 발달 수준을 보정한 값이다.

둘째, 소프트웨어 개발자의 전문성 발달 수준에 따라 5개의 집단으로 구분되었다. Cluster 1은 전문성 발달 수준 평균 1.60이었으며, 초급개발자로 명명하였고, Cluster 2는 평균 2.38로 중급개발자, Cluster 3은 평균 2.97로 고급개발자, Cluster 4는 평균 3.47로 특급개발자, Cluster 5는 평균 4.13으로 개발전문가로 명명하였다. 교육훈련, 자격 및 직무경력을 토대로 등급별 특성을 살펴보면 Cluster 1(초급개발자)은 노임단가 초급기술자 수준이었고 전문대학에서 4년제 대학의 중간 수준(2.89)이었다. 입직 후 상급학교 진학은 5.3%, 소프트웨어 관련 훈련에 평균 5.85개, 비관련 훈련에 평균 9.5개 참여하였다. 평균 기사 자격을 소지하고 있으며, 7개 프로젝트에 참여하고 5.37년 재직하는 것으로 확인되었다. Cluster 2(중급개발자)는 노임단가 중급기술자 수준이었으며 4년제 대학(2.96)수준이었다. 입직 후 상급학교 진학은 8.4%, 소프트웨어 관련 훈련은 평균 2.34개, 비관련 훈련에 3.37개 참여하였다. 평균 기사 자격을 소지하고 있으며, 9.37개 프로젝트에 참여하고 7.40년 재직하는 것으로 확인되었다. Cluster 3(고급개발자)는 노임단가 중급기술자 수준이었으며, 4년제 대학(2.97)수준이었다. 입직 후 상급학교 진학은 11.8%, 소프트웨어 관련 훈련은 평균 3.26개, 비관련 훈련은 평균 11.57개 참여하였다. 평균 기사 자격 수준이었으며,

11.45개 프로젝트에 참여, 8.53년 재직한 것으로 확인되었다. Cluster 4(특급개발자)는 노임단가 중급기술자 수준이었으며 4년제 대학(3.03)수준이었다. 입직 후 상급학교 진학은 7.0%, 소프트웨어 관련 훈련은 5.08개, 비관련 훈련은 13.3개 참여하였다. 평균 기사 자격 수준이었으며, 18.03개 프로젝트 참여, 10.71년 재직한 것으로 확인되었다. Cluster 5(개발 전문가)는 노임단가 고급기술자 수준이었으며 4년제 대학에서 석사학위(3.12) 수준이었다. 입직 후 상급학교 진학은 9.1%, 소프트웨어 관련 훈련 9.73개, 비관련 훈련에 44개 참여하였다. 평균 기사 자격 수준이었으며, 49.03개 프로젝트 참여 11.18년 재직한 것으로 확인되었다.

셋째, 전문성 발달 수준과 교육훈련(학교교육, 평생교육), 자격(자격증 보유 자격, 프로젝트 수행 자격)의 상관관계는 프로젝트 수행 자격, 직무경력, 자격증 소지 자격, 학교교육 순으로 상관관계가 높게 나타났으며, 평생교육은 통계적으로 유의미한 관계가 없는 것으로 나타났다. 전문성 발달 수준과 교육훈련, 자격 및 직무경력 변인 중 프로젝트 수행 자격 .181, 직무경력 .151, 자격증 소지 자격 .146, 학교교육 .120이 통계적으로 유의미하게 전문성 발달 수준에 영향을 미치는 것으로 나타났다. 반면 전문성 발달 수준과 노임단가 기준 변인간의 관계를 분석한 결과는 노임단가 기준 변인 중에서는 종사년수, 최종학력순으로 상관관계가 높게 나타났으며, 최고 수준 소지 자격증은 상관이 없는 것으로 나타났다. 전문성 발달 수준과 노임단가 기준 변인 중 종사년수 .211, 최종학력 .099가 통계적으로 유의미하게 전문성 발달 수준에 영향을 미치는 것으로 나타났다.

넷째, 교육훈련, 자격 및 직무경력을 활용한 모형에서 효력지수에 의한 변인의 중요도 순서는 프로젝트 수행 자격(0.765)>직무경력(0.753)>학교교육(0.184)>자격증 소지 자격(0.109)>평생교육(0.103)의 순으로 나타났다. 또한, 교육훈련, 자격 및 직무경력을 활용한 모형을 활용한 경우 전문성 등급과의 분류 정확률이 43.5%로 기존 노임단가 변인을 활용한 경우의 분류 정확률인 34.1%보다 높게 나타났다. 소프트웨어 개발자의 전문성 등급을 구분하는 관련 변인들의 판별력을 구명하기 위해 판별분석을 한 결과, 노임단가 기준 모형에서 효력지수에 의한 변인의 중요도 순서는 최종학력(0.624)>보유자격증의 최고 수준(0.353)>종사년수(0.190)의 순으로 나타났다.

이 연구의 결론을 종합적으로 다음과 같이 제시하였다. 첫째, 소프트웨어 개발자의 전문성 발달 수준은 개발 생명주기(life cycle)를 반영한 분석 및 설계, 구현, 테스트, 프로젝트 관리로 구성된 측정도구로 측정할 수 있다. 둘째, 소프트웨어 개발자의 전문성 발달 수준에 따른 등급은 초급개발자, 중급개발자, 고급개발자, 특급개발자, 개발전문가 5단계로 설정할 수 있었고, 초급개발자에서 개발전문가로 갈수록 전문성 발달 수준이 높아졌다. 셋째, 소프트웨어

개발자의 전문성 발달 수준을 예측하기 위해 교육훈련, 자격 및 직무경력의 대리지표를 사용할 때에는 최종학력, 보유자격증의 최고 수준, 종사년수로 단편적인 측정을 하는 것이 아니라 학력과 전공을 고려한 학교교육, 입직 후 상급학교 진학과 사내외 훈련을 고려한 평생교육, 자격증의 보유 수준, 개수, 유형을 고려한 자격증 소지 자격, 직급별 참여한 프로젝트 개수를 고려한 프로젝트 수행 자격과 직급과 직급별 재직년수를 고려한 직무경력으로 측정해야 할 것이다. 넷째, 전문성 등급을 구분짓는데 가장 중요한 변인은 프로젝트 수행 자격과 직무경력이었다. 이 결과는 그동안 전문성을 예측함에 있어 학력과 자격증을 중요시 한 기존 사회현상을 반증하는 결과였다. 한 개인의 교육훈련, 자격 및 직무경력을 측정함에 있어 편의성만 고려하여 간소화한 방법으로 측정한 변인으로는 전문성을 정확하게 예측하지 못했다는 것이 반증되었으며, 가장 중요한 변인이었음에도 개인이 현장에서 쌓아온 다양한 경험들이 그동안 간과되었다는 사실을 알 수 있었다. 이는 최근에 직업교육분야에서 관심을 받고 있는 선행학습경험(RPL), 능력중심사회와 연계될 수 있을 것이다.

이 연구 결과를 바탕으로 후속연구를 위한 제언을 하면 다음과 같다. 첫째, 소프트웨어 개발자의 전문성 수준을 면밀하게 측정하게 측정하기 위해서는 경력단계별 구인가중치를 다르게 적용해야 한다. 둘째, 교육훈련, 자격 및 직무경력 변인을 관찰가능한 다양한 요소들을 포함하여 측정하고자 시도하였지만 추가적인 지표들을 도출하여 더 정확하게 측정해야 한다. 셋째, 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계를 분석하는 경우에는 개인의 현재 직무나 근무하고 있는 기업 규모도 고려해서 통제해서 분석할 필요가 있다. 넷째, 전문성 등급과 교육훈련, 자격 및 직무경력의 차이분석을 통해 등급별 교육훈련, 자격 및 직무경력의 유의미한 차이를 분석할 필요도 있다. 다섯째, 전문성 발달 수준에 영향을 미치는 결정요인으로 교육훈련, 자격 및 직무경력을 선정하였지만, 인지심리학 및 교육학 분야에서는 객관적인 교육훈련, 자격 및 직무경력에 대한 투입요소과 함께 개인이 얼마나 의지를 가지고 충실하게 참여하는지에 대한 의도적인 연습(deliberate practice)의 매개효과를 보는 것도 필요할 것이다.

주요어 : 소프트웨어 개발자, 전문성 발달 수준, 전문성 등급, 교육훈련, 자격, 직무경력

학 번 : 2011-23466

목 차

I. 서론	1
1. 연구의 필요성	1
2. 연구목적	3
3. 연구문제	3
4. 용어의 정의	5
5. 연구의 제한	7
II. 이론적 배경	8
1. 소프트웨어 산업과 개발자	8
2. 전문성의 개념과 측정	15
3. 전문성 발달과 경력경로	42
4. 전문성과 교육훈련, 자격 및 직무경력 관계	54
III. 연구 방법	71
1. 연구모형	71
2. 연구대상	71
3. 조사도구 및 변인 측정	73
4. 자료수집	95
5. 자료분석	96

IV. 연구 결과100

1. 소프트웨어 개발자의 일반적 특성 및 변인 측정100
2. 소프트웨어 개발자의 전문성 발달 수준과 등급114
3. 소프트웨어 개발자의 전문성 등급별 특성125
4. 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계130
5. 교육훈련, 자격 및 직무경력 변인의 전문성 등급 결정 크기134
6. 연구 결과에 대한 논의139

V. 요약, 결론 및 제언146

1. 요약146
2. 결론149
3. 제언150

참고문헌152

[부록 1] 설문도구165

[부록 2] 측정도구의 예비조사 분석 결과172

[부록 3] 전문성 등급에 따른 관련 변인의 차이분석 결과174

표 차례

<표 II-1> 한국표준산업분류 소프트웨어 관련 산업	9
<표 II-2> 소프트웨어 개발자의 정의	12
<표 II-3> 한국표준직업분류에 따른 소프트웨어 개발 관련 직업과 직무의 관계	14
<표 II-4> 전문성에 대한 관점별 주요 전문성 정의 예시	16
<표 II-5> 전문성 관련 선행연구에서 제시한 구성요소	26
<표 II-6> 일반적인 전문성 구성요소로 본 소프트웨어 개발자의 전문성 관련 주요 선행연구	28
<표 II-7> Sonnentag(1995)의 소프트웨어 전문성 요소	29
<표 II-8> Sonnentag(1998)의 소프트웨어 전문성 요소	30
<표 II-9> Sonnentag(2006)의 소프트웨어 전문성 관련 유경험자, 고성과자의 행동 비교	31
<표 II-10> Wynekoop 외 (2000) 소프트웨어 전문가의 측정가능한 특성요소	32
<표 II-11> Chilton & Hardgrave(2004)의 IT 성과 측정 요소	33
<표 II-12> Prasad 외(2010)의 소프트웨어 전문성의 성과 지표	34
<표 II-13> 국가별 소프트웨어 표준을 통해 고찰한 전문성 구성요소	35
<표 II-14> 한국의 TOPCIT(2014)의 구성요소	36
<표 II-15> 전문성 측정 방법 장단점 비교	40
<표 II-16> 전문성의 발달 과정을 본 선행연구	44
<표 II-17> 소프트웨어 산업 기술자의 등급 분류 기준표	47
<표 II-18> 선행연구에서의 교육훈련 측정방법	59
<표 II-19> 자격의 개념별 활용과 예시	62
<표 II-20> 선행연구에서의 자격 측정방법	64
<표 II-21> 선행연구에서의 직무경력 측정방법	70

<표 III-1> 전문성 발달 5단계에 근거한 전문성 발달 수준 척도	76
<표 III-2> 도구 검토 전문가	76
<표 III-3> 소프트웨어 개발자의 전문성 발달에 대한 확인적 요인 분석 결과	78
<표 III-4> 학교급별 전공유무 유형에 따른 상대적 중요도	82
<표 III-5> 소프트웨어 개발자의 소프트웨어 관련 훈련시간 변환 기준	85
<표 III-6> 소프트웨어 개발자의 소프트웨어 관련 비관련 훈련시간 변환 기준	85
<표 III-7> 소프트웨어 개발 관련 국가기술자격 등급별 자격증의 수	88
<표 III-8> 국가공인민간자격과 해외자격에 대한 가중치 변환 기준	88
<표 III-9> 자격증 중복 유형 따른 상대적 중요도	89
<표 III-10> 소프트웨어 개발자의 직급별 프로젝트 참여 변환 기준	92
<표 III-11> 소프트웨어 개발자의 소프트웨어 종사년수 변환 기준	94
<표 III-12> 통계분석방법	96
<표 III-13> 전문성 등급을 구분하는 판별모형	98
<표 IV-1> 응답한 소프트웨어 개발자의 입직 시 학력	100
<표 IV-2> 응답한 소프트웨어 개발자의 현재 최종학력	101
<표 IV-3> 응답한 소프트웨어 개발자의 입직 시 학교급별 전공	102
<표 IV-4> 응답한 소프트웨어 개발자의 현재 학교급별 전공	102
<표 IV-5> 소프트웨어 개발자의 학교교육 점수	103
<표 IV-6> 입직 시와 현재 대비 학력 변화 빈도	104
<표 IV-7> 응답한 소프트웨어 개발자의 입직 후 진학 학교에서의 전공	104
<표 IV-8> 소프트웨어 관련 훈련 개수 및 시간(결측치 포함)	105
<표 IV-9> 소프트웨어 관련 훈련 개수 및 시간(결측치 제외)	106
<표 IV-10> 입직 시 학력별 소프트웨어 훈련 프로그램 참여 정도	106
<표 IV-11> 소프트웨어 개발자의 평생교육 점수	107
<표 IV-12> 입직 시 학교급별 평생교육 점수	108

<표 IV-13> 응답한 소프트웨어 개발자의 최고 자격증 수준	109
<표 IV-14> 소프트웨어 개발 자격 중복 유형별 응답한 개발자 비율	110
<표 IV-15> 응답한 소프트웨어 개발자의 수준별 소지 자격 수 빈도	111
<표 IV-16> 소프트웨어 개발자의 자격증 소지 자격 점수	111
<표 IV-17> 응답한 소프트웨어 개발자가 직급별 프로젝트 참여 개수	112
<표 IV-18> 소프트웨어 개발자의 프로젝트 수행 자격 점수	113
<표 IV-19> 응답한 소프트웨어 개발자의 총 경력기간	113
<표 IV-20> 응답한 소프트웨어 개발자의 직급별 경력기간	114
<표 IV-21> 응답한 소프트웨어 개발자의 분야별 경력기간	114
<표 IV-22> 소프트웨어 개발자의 직무경력 점수	115
<표 IV-23> 소프트웨어 개발자의 전문성 발달 수준 측정도구에 대한 문항별 반응 분포	117
<표 IV-24> 전문성 발달 수준 측정도구에 응답한 값	120
<표 IV-25> 전문성 발달 구성요소의 주성분 분석 결과	120
<표 IV-26> 가중치로 보정한 소프트웨어 개발자의 전문성 발달 수준	121
<표 IV-27> 전문성 발달 수준에 따른 군집분석 결과	122
<표 IV-28> 전문성 등급과 이론적 등급의 교차분석 결과	123
<표 IV-29> 응답한 소프트웨어 개발자의 전문성 등급의 군집분석 유의도 검사	124
<표 IV-30> 군집분석 결과에 따른 전문성 등급	126
<표 IV-31> 전문성 등급별 특성 종합	130
<표 IV-32> 교육훈련, 자격 및 직무경력과 전문성 발달 수준의 상관관계	132
<표 IV-33> 교육훈련, 자격 및 직무경력과 전문성 발달 수준의 타당성 확보	133
<표 IV-34> 전문성 발달 수준과 교육훈련, 자격 및 직무경력간의 회귀분석 결과	133
<표 IV-35> 노임단가 기준 변인과 전문성 발달 수준의 상관관계	134
<표 IV-36> 노임단가 기준 변인과 전문성 발달 수준의 타당성 확보	135
<표 IV-37> 전문성 발달 수준에 대한 노임단가 기준 변인의 회귀분석	135
<표 IV-38> 교육훈련, 자격 및 직무경력 판별모형 동일성 가정에 대한 Box' M값	136

<표 IV-39> 교육훈련, 자격 및 직무경력 판별모형의 고유값 및 윌크스람다값	137
<표 IV-40> 교육훈련, 자격 및 직무경력을 활용한 판별함수 결과 요약	138
<표 IV-41> 전문성 등급과 교육훈련, 자격 및 직무경력 모형에 따른 예측집단 분류율	138
<표 IV-42> 노임단가 기준 변인 판별모형 동일성 검증에 대한 Box' M값	139
<표 IV-43> 노임단가 기준 변인 판별모형의 고유값 및 윌크스람다값	139
<표 IV-44> 노임단가 기준 변인 판별함수 결과 요약	140
<표 IV-45> 전문성 등급과 노임단가 기준 변인 모형에 따른 예측집단 분류율	141

그림 차례

[그림 II-1] 전문성 연구의 새로운 흐름도	45
[그림 II-2] 역량유형 × 역량수준 매트릭스	46
[그림 II-3] 소프트웨어 종사인력의 경력경로	49
[그림 II-4] 소프트웨어 전문인력(기술개발직군)의 경력경로	50
[그림 II-5] 직무경력 측정의 개념적 틀	67
[그림 III-1] 소프트웨어 개발자의 전문성 발달 수준과 교육훈련 자격 및 직무경력의 관계 연구모형	72
[그림 III-2] 소프트웨어 개발자의 전문성 구인	74
[그림 IV-1] 군집별 빈도에 대한 정규성 검증	124

I. 서론

1. 연구의 필요성

IT를 기반으로 정보가 넘쳐나고 빠르게 변해가는 지식사회는 이전 산업사회와 다른 인력을 요구한다. 지식기반사회에서는 고도의 지식, 기술, 그 외 정보이용능력, 창의성 등의 역량을 활용하여 주어진 맥락과 상황에 맞게 문제를 해결해나가는 전문성이 요구되기 시작했다(배을규, 동미정, 이호진, 2011; 오현석, 2009). 따라서 국가, 사회의 경쟁력을 이끌어나가는 사람은 단순히 일을 잘하는 보통 수준의 다수가 아니라 특정 분야에서 뛰어난 능력을 갖춘 전문성을 겸비한 사람으로 변화되었다(오현석 외, 2007; 조동성, 문휘창, 2006). 그러나 분야를 불문하고 많은 사람들이 경력의 첫 단계에서 요구하는 수준에는 쉽게 도달하지만, 한 분야의 최고 수준까지는 도달하지 못하고 있어 최고 수준의 전문성을 갖춘 고급인력의 수급 문제가 분야를 막론하고 지속적으로 제기되고 있다.

IT 산업은 국가마다 미래를 이끄는 산업으로 지목되면서 IT 산업의 핵심인력인 소프트웨어 개발자의 전문성 문제는 개인의 문제를 넘어 조직, 사회, 국가 차원에서 관심이 확대되었다(이상돈 외, 2012). 또한 IT 산업은 한 개인이 보유한 전문성 수준에 따라 10,000배 이상의 성과 차이가 나는 산업으로 개인의 능력을 높이기 위한 다양한 노력이 시도되었다. EU 국가, 일본, 미국, 영국 등 각 국가에서는 소프트웨어 분야 인력의 전문성 발달을 위한 국가 차원의 프레임워크를 구축하여 체계적으로 관리하려는 노력을 진행하였다. 하지만 아직까지 우리나라는 개발자들의 전문성 보유 수준에 대한 체계적인 관리가 미흡하며, 개발자에 대한 처우도 열악하여 일정 수준의 전문성을 갖춘 고급인력이 미국, 유럽 등의 선진국으로 유출되고 있어 이들에 대한 체계적인 관리가 필요한 상황이다(위재경, 2015.04.21.).

지금까지 전문성 연구는 행동심리학 이후 인지심리학이 등장하면서 최고 수준의 체스(chess) 챔피언과 보통 사람들의 인지과정 차이를 비교하는 연구에서 시작하여, 다양한 학문 분야로 확대되었다(Ericsson & Kintsch, 2000). 전문성을 정의함에 있어 하나의 합의된 개념으로 정의하기는 어렵지만(Mieg, 2007), 일반적으로 전문성은 특정 분야에서 최고 수준으로 인정받는 전문가들이 보이는 특성으로 정의되고 있으며(Harmon & King, 1986), 다양한 경험과 학습을 통해 지속적으로 발달되어야 하는 과정으로 보는 것이 일반적이다(Mieg, 2007). 전문성을 보는 관점은 학문분야에 따라 상이한데, 개인의 인지적, 기질적 특성으로 보는 인지심리학적 관점(Ericsson & Lehnamm, 1999; Kochevar, 1994; Harmon &

King, 1986), 생산성을 높이기 위한 인적자본으로 보는 사회경제학적 관점(Mieg, 2007; Feltovich, Prietula, & Ericsson, 2007) 그리고 학습을 통한 개인의 역량으로 보는 HRD 관점(Nunn, 2008; Benner, 1986; Dreyfus & Dreyfus, 1982) 등으로 구분될 수 있다.

이들 분야에서 공통적으로 관심을 두고 있는 것은 전문성을 객관적으로 측정하고, 전문성을 발달시키기 위한 다양한 요인을 탐색하는 것이었다. 하지만 전문성을 측정하는 방법에 있어 실험실이라는 제한된 공간에서 몇 개의 인지테스트로 전문성을 측정하거나(Ericsson & Ward, 2007), 전문성 보유 수준에 대한 자가측정(Mieg, 2007)의 한계를 넘기 위해 다른 동료들의 추천을 통해 상대적인 수준을 파악하는(Sonnentag, 1995, 1998) 방법들을 활용하였으나 전문성을 명확하게 측정하였다고 보기에는 한계가 있었다. 또한, 질적연구는 다수가 특정 개인의 일대기를 다루거나 성공신화적인 측면이 부각되어 전문성 일반화 및 객관적 결과를 얻을 수 없었다(손은정, 유성경, 심혜원, 2003; 윤형한, 2009; 신중호 외, 2007; 김정아, 2007; 오현석, 2007; 한상만, 이희수, 2014; 이재실, 2011). 한편, 전문성의 발달 단계에 대한 이론들은 점차 확립되어 가고 있음에도(Dreyfus & Dreyfus, 1977; Benner, 1982) 소프트웨어 개발자를 대상으로 한 단계별 수준의 차이가 균등한지, 단계마다 머물게 되는 소유기간이 일정한지 등 발달과정에 대한 구체적인 설명이 미흡하여 전문성 발달의 측정 및 발달 과정에 대한 객관적인 연구 결과에 한계가 있었다.

더욱이 전문성 예측하는 대리지표로 교육훈련, 자격 및 직무경력을 주로 활용하였다. 교육훈련을 측정함에 있어서도 단순히 학력 또는 학벌로 평가하였으며(Chi, Claser & Farr, 1988; 김난영 & 조원혁), 개인이 보유한 자격수준을 검증할 수 있는 지표가 다양함에도 자격증에만 의존하여 평가하는 등 측정의 편이성만을 고려하였다. 또한, 학령기 시절 이후 교육훈련을 받지 않던 산업사회에 비해 100세 시대, 평생학습시대를 맞이하여 성인학습자의 비중이 크게 늘어났으며 정규교육 이외 다양한 학습과정이 등장하면서 다차원적인 학습이 가능하게 되었음에도(변정현, 2011; 오석영, 2011) 평생교육에 대한 연구가 활발하지 않았다. 개인이 쌓아가는 경력에 대해서도 여러 연구를 통해 경력의 기간이 일정 수준의 전문성 발달 수준에 도달하는데 영향을 미치지만 일정 수준을 넘어가기 위해서는 직무경험을 양적인 접근 외 질적인 접근과 함께 이루어야 한다는 연구 결과가 제기되었음에도(Quinones, Ford & Teachou, 1995; Tesluk & Jacobs, 1998), 직무경력을 측정함에 있어서도 전체 경력기간만을 활용하는 한계가 나타났다.

이와 같은 내용을 종합해볼 때, 소프트웨어 개발자들의 전문성에 대한 연구는 중요한 의의를 가진다. 우선 소프트웨어 개발자의 전문성을 객관적으로 측정할 수 있는 도구를 개발하고, 이에 대한 전문성 발달 수준 및 발달 수준에 따른 등급을 구분하는 연구가 필요하다. 소프트

웨어 개발자의 전문성 발달에 영향을 미치는 교육훈련, 자격 및 직무경력을 보다 정밀하게 측정하는 방안을 모색하고, 전문성 발달 수준과의 관계를 분석해볼 필요가 있다. 특히 소프트웨어 분야의 경우 소프트웨어 개발 전문성을 객관적으로 측정하는 대신 학력, 국가기술자격 등의 보유 수준, 종사기간만으로 소프트웨어 산업 기술자들의 노임대가를 등급으로 구분하고 있었으나, 이에 대한 타당성이 미흡하여 현장의 불만이 지속적으로 제기되었다. 따라서 기존 노임단가등급에서 활용하고 있는 교육훈련, 자격 및 직무경력의 변인과 이 연구에서 새롭게 측정한 교육훈련, 자격 및 직무경력 중 어느 측정 방법이 전문성의 발달에 영향을 미치는지 확인한다는 데 큰 의의가 있을 것이다.

2. 연구목적

이 연구의 목적은 소프트웨어 개발자의 전문성 발달수준과 교육훈련, 자격 및 직무경력의 관계를 구명하는데 있다. 이를 달성하기 위해 다음과 같이 세부 연구목표를 설정하였다.

첫째, 소프트웨어 개발자의 전문성의 발달 수준을 구명한다.

둘째, 소프트웨어 개발자의 전문성 발달 수준에 따라 등급을 구분하고 등급별 특성을 구명한다.

셋째, 소프트웨어 개발자의 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계를 구명한다.

넷째, 소프트웨어 개발자의 전문성 등급 집단을 구분하는 교육훈련, 자격 및 직무경력의 판별력을 구명한다.

3. 연구문제

연구 목적 달성을 위해 다음과 같은 연구 문제를 설정하였다.

연구문제 1. 소프트웨어 개발자의 전문성 발달 수준은 어떠한가?

1-1. 소프트웨어 개발자의 전문성 발달을 구성하는 요인은 무엇인가?

1-2. 소프트웨어 개발자의 전문성 발달을 구성하는 요인별 가중치로 보정한 전문성 발달 수준은 어떠한가?

연구문제 2. 소프트웨어 개발자의 전문성 발달 수준에 따른 전문성 등급은 어떻게 구분될 수 있는가? 등급별 특성은 어떠한가?

2-1. 소프트웨어 개발자의 전문성 등급은 5개(Dreyfus & Dreyfus, 1980)로 구분되는가?

2-2. 소프트웨어 개발자의 전문성 등급별 교육훈련, 자격 및 직무경력의 차이는 어떠한가?

연구문제 3. 소프트웨어 개발자의 전문성 발달 수준과 교육훈련, 자격 및 직무경력 및 노임단가 기준 변인의 관계는 어떠한가?

3-1. 교육훈련(학교교육, 평생교육), 자격(자격증 소지 자격, 프로젝트 수행 자격) 및 직무경력과 전문성 발달 수준은 정적인 관계인가? 설명력은 어떠한가?

3-2. 노임단가 기준 변인인 최종학력, 보유 자격증의 최고 수준 및 종사년수와 전문성 발달 수준은 정적인 관계인가? 설명력은 어떠한가?

3-3. 교육훈련, 자격 및 직무경력의 관계와 노임단가 기준 변인의 관계에는 어떠한 차이가 있는가?

연구문제 4. 소프트웨어 개발자의 전문성 등급 집단을 구분하는 교육훈련, 자격 및 직무경력의 판별력은 어떠한가?

4-1. 소프트웨어 개발자의 전문성 등급 집단을 구분하기 위한 교육훈련(학교교육, 평생교육), 자격(자격증 소지 자격, 프로젝트 수행 자격), 직무경력의 판별력은 어떠한가?

4-2. 소프트웨어 개발자의 전문성 등급 집단을 구분하기 위한 노임단가 기준 변인인 최종학력, 보유 자격증의 최고 수준 및 종사년수의 판별력은 어떠한가?

4-3. 교육훈련, 자격 및 직무경력과 노임단가 기준 변인의 판별력에는 어떠한 차이가 있는가?

4. 용어의 정의

가. 소프트웨어 개발자

소프트웨어 개발자란 소프트웨어 프로젝트 또는 제품을 개발하는 일련의 과정을 수행하는 자로(Wynekoop & Walz, 2000), 구현(programming) 업무만 수행하는 프로그래머보다는 더 넓은 개념이다. 따라서 이 연구에서는 소프트웨어 개발자를 소프트웨어 개발 생명주기(Life Cycle) 과정인 분석 및 설계, 구현 및 테스트와 전체 프로젝트를 총괄하는 관리 업무를 수행하는 자로 정의한다.

나. 소프트웨어 개발자의 전문성

일반적으로 전문성이란 특정 분야에서 최고 수준의 전문가가 지닌 지식, 경험, 기술, 창의성, 직관력 등의 능력(Harmon & King, 1986; Kochevar, 1994)을 의미한다. 이 연구에서 소프트웨어 개발자의 전문성이란 최고 수준의 소프트웨어 개발자가 지닌 분석 및 설계, 구현, 테스트 및 프로젝트 관리 능력을 의미한다.

다. 전문성 발달

전문성 발달이란 최고 수준의 전문가가 지닌 능력의 수준까지 도달해가는 과정을 의미하며, Dreyfus & Dreyfus(1980)는 초보자(novice), 고급입문자(advanced beginner), 능숙자(competent), 숙련자(proficient), 전문가(expert)의 5단계로 발달 과정을 거친다고 보았다.

따라서 소프트웨어 개발자의 전문성 발달 수준이란 최고 수준의 소프트웨어 개발자가 지닌 분석 및 설계, 구현, 테스트 및 프로젝트 관리 수준에 도달한 정도를 의미한다. 이 연구에서는 Dreyfus & Dreyfus(1980)이 제시한 전문성 발달 단계별 특성을 토대로 연구자가 5점 리커트 척도로 변환하여 개발한 도구에 응답한 값을 기초로 분석 및 설계, 구현, 테스트 및 프로젝트 관리의 상대적 중요도에 따른 가중치를 적용하여 변환한 값을 의미한다.

라. 전문성 등급

전문성 등급은 전문성 발달 구성요소별 가중치로 보정한 값인 전문성 발달 수준 점수를 가지고 5개 집단(Dreyfus & Dreyfus, 1980)으로 구분한 집단들을 의미한다.

마. 교육훈련

교육훈련은 일반적으로 능력을 함양하기 위해 이수한 학습을 의미하며, 이 연구에서는 입직 전 학령기에 정규교육을 통해 이수한 학교교육과 개인이 입직 후 자발적으로 참여한 평생교육으로 구분된다.

1) 학교교육

학교교육은 정규교육과정에서 이수한 수준을 의미하며, 일반적으로는 개인이 취득한 최종 졸업장 및 학위인 최종학력을 의미한다. 이 연구에서는 입직 전 이수한 정규교육을 학교교육으로 보고, 최종학력과 학교급별 소프트웨어 개발 전공을 고려해서 계산한 값으로 정의한다.

2) 평생교육

평생교육은 태어나서 요람까지 일생동안 이루어지는 교육을 의미하며 일반적으로 정규교육이 아닌 형태로 이루어지는 교육을 의미한다. 이 연구에서는 입직 이후 개인이 참여한 정규교육 외 훈련과 입직 이후 상급학교에 진학한 경우까지 평생교육으로 보고, 입직 후 상급학교의 진학과 입직 후 사내외 훈련 및 연수를 고려해서 계산한 값으로 정의한다.

바. 자격

자격은 특정 직업에서 직무를 수행하는데 필요한 능력을 의미하며, 해당 능력이 일정 수준에 도달했다는 것은 검증된 결과인 자격증, 인증서 등으로 확인할 수도 있으나 검증된 자료가 없음에도 그동안의 경험을 토대로 이에 준하는 능력 수준을 보유한다고 인정받을 수 있다. 따라서 이 연구에서는 자격을 자격증 소지 자격과 소프트웨어 분야에서 자격증에 준하는 경험으로 인정받는 프로젝트 수행 자격으로 자격을 구분하였다.

1) 자격증 소지 자격

자격증 소지 자격은 개인이 보유한 능력을 검증된 절차를 통해 인정받은 결과를 의미한다. 일반적으로 자격증 소지는 개인이 최고 수준 소지 자격증으로 측정하나 이 연구에서는 최고 수준 자격 등급, 자격 보유 개수와 자격 보유 유형을 고려해서 계산한 값을 의미한다.

2) 프로젝트 수행 자격

프로젝트 수행 개인이 보유한 능력에 대해 검증되지는 않았으나 프로젝트를 수행한 경우 관련 능력을 일정 수준 보유하고 있다고 인정될 수 있기 때문에 자판단하는 보유하고 있다고 판단한도 수행 자격으로 정의된다. 이 연구에서는 직급별 참여한 프로젝트의 개수를 고려해서 계산한 값을 의미한다.

사. 직무경력

직무경력에 입직 후 직무를 수행한 경험의 총체를 의미하며, 일반적으로 총 종사년수를 의미한다. 이 연구에서는 직무경력을 입직 후 직무를 통해 수행한 경험의 총체로 정의하며, 개인이 최고 도달한 직급과 직급별 종사한 기간을 고려해서 계산한 값을 의미한다.

5. 연구의 제한

이 연구는 소프트웨어 개발자를 대상으로 소프트웨어 개발 분야에 한정된 연구이다. 따라서 이 연구의 결과는 소프트웨어 개발자에게는 적합하지만 다른 산업으로 일반화하기에는 한계가 있다.

II. 이론적 배경

1. 소프트웨어 산업과 개발자

가. 소프트웨어 산업의 범위

한국표준산업분류(KSIC) 9차 개정에 따르면 소프트웨어 산업은 ‘출판, 영상, 방송통신 및 정보서비스업(J)’에 속한다. 출판, 영상, 방송통신 및 정보서비스업(J)은 ‘정보 및 문화상품을 생산하거나 공급하는 산업 활동; 정보 및 문화상품을 전송하거나 공급하는 수단을 제공하는 산업 활동; 통신서비스 활동; 정보기술, 자료처리 및 기타 정보서비스를 제공하는 산업활동’으로 정의된다. 일반적으로 출판업, 소프트웨어 제작, 영상 및 오디오 기록 활동, 라디오 및 텔레비전 방송, 프로그램공급, 전기통신, 정보기술 및 기타 정보서비스 활동이 이에 포함된다고 볼 수 있다. 이 중 소프트웨어 산업은 컴퓨터 시스템의 통합관련 기획 및 설계서비스를 주로 제공하는 산업활동과 컴퓨터 시스템의 관리 및 운영관련 기술서비스를 주로 제공하는 산업 활동으로 구분한 ‘컴퓨터 프로그래밍, 시스템 통합 및 관리업’에 속한다(통계청, 2014).

출판, 영상, 방송통신 및 정보서비스업(J) 분야 중 컴퓨터 프로그래밍, 시스템 통합 및 관리업과 관련된 세부 산업은 게임 소프트웨어 개발 및 공급업, 시스템·응용 소프트웨어 개발 및 공급업이 포함된 소프트웨어 개발 및 공급업, 컴퓨터 프로그래밍 서비스업, 컴퓨터시스템 통합 자문, 구축 및 관리업, 기타 정보기술 및 컴퓨터운영 관련 서비스업이 포함된 컴퓨터 프로그래밍, 시스템 통합 및 관리업, 자료처리, 호스팅 및 관련 서비스업, 포털 및 기타 인터넷 정보매개 서비스업이 포함된 자료처리, 호스팅, 포털 및 기타 인터넷 정보매개서비스업, 기타 정보서비스업으로 구분된다(<표 II-1> 참고).

소프트웨어산업진흥법 제2조(정의)1(2010.03.31. 일부개정)에 따르면 소프트웨어(이하, 소프트웨어)란 ‘컴퓨터, 통신, 자동화 등의 장비와 그 주변 장치에 대하여 명령, 입력, 처리, 저장, 출력, 상호작용이 가능하도록 하게 하는 지식, 명령(음성이나 영상정보 포함)의 집합과 이를 작성하기 위하여 사용된 기술서 기타 관련 자료’를 의미하며 통상 OS와 같은 시스템을 운영 및 관리하는 ‘시스템 소프트웨어’, 프로그램을 개발 및 설계하는 ‘개발용 소프트웨어’, 임의의 응용 분야에서 특정 목적을 수행하는 ‘응용 소프트웨어’로 구분한다(한국소프트웨어진흥협회, 2014). 위와 동령 제2조(정의)2(2010.03.31. 일부개정)에 의거하면 소프트웨어산업이란 소프트웨어개발, 제작, 생산, 유통 등과 이에 관련된 서비스 및 정보시스

템의 구축, 운영 등과 관련된 산업으로 정의하였다. 따라서 정보통신산업진흥원(2011)은 소프트웨어산업을 소프트웨어개발 단계부터 최종소비자(기업 또는 개인)가 사용하기까지 소프트웨어공급과 관련된 모든 생산적 활동 전반을 포함하는 영역으로 전통적으로 패키지소프트웨어, IT 서비스, 임베디드 소프트웨어 등 3개 분야로 독립형(패키지 소프트웨어)이자, 중간재(임베디드 소프트웨어)이며, 서비스형(IT 서비스)산업으로 구분했다.

〈표 II-1〉 한국표준산업분류 소프트웨어 관련 산업

코드	항목명
582	소프트웨어 개발 및 공급업
5821	게임 소프트웨어 개발 및 공급업
5822	시스템·응용 소프트웨어 개발 및 공급업
620	컴퓨터 프로그래밍, 시스템 통합 및 관리업
6201	컴퓨터 프로그래밍 서비스업
6202	컴퓨터시스템 통합 자문, 구축 및 관리업
6209	기타 정보기술 및 컴퓨터운영 관련 서비스업
631	자료처리, 호스팅, 포털 및 기타 인터넷 정보매개서비스업
6311	자료처리, 호스팅 및 관련 서비스업
6312	포털 및 기타 인터넷 정보매개 서비스업
639	기타 정보 서비스업
6391	뉴스 제공업

주: 통계청(2014). 한국표준산업분류 실무 적용을 위한 분류 설명서.

이에 따라 이 연구에서는 한국표준산업분류와 소프트웨어산업진흥법을 고려하여 정보통신산업진흥원(2011)이 정의한 소프트웨어개발 단계부터 최종소비자(기업 또는 개인)가 사용하기까지 소프트웨어공급과 관련된 모든 생산적 활동 전반을 포함하는 영역으로 정의한다.

우리나라를 포함한 선진국들은 지식상품, 지식근로자, 지식기술근로자로부터 생산성을 증대시킬 수 있는 새로운 미래 산업에 우선권을 두기 시작했다(Drucker, 1997). 특히 우리나라의 경우, 한국 경제의 차세대 산업으로 소프트웨어 산업을 꼽고 미래 핵심 산업 및 국가 기간산업으로 성장시키겠다는 전략을 제시하였다. 이로 인해 소프트웨어 산업인력을 정보화 시대를 주도하는 주요 인력으로 강조하고, 국가차원에서도 관련 인력을 효율적으로 양성하기 위한 노력을 기울이고 있는 실정이다(미래창조과학부, 2014).

나. 소프트웨어 개발자의 정의와 직무

1) 소프트웨어 개발자의 정의

소프트웨어 개발자는 영문권에서는 Software Developer(Wynekoop & Walz, 2000) 또는 Software Designer, Programmer(Sonnentag, Niessen & Volmer, 2007)등의 용어로 혼용되어 사용되고 있으며 우리나라에서도 개발자, 프로그래머, 분석가 및 설계자 등 연구별로 다른 기준과 용어를 사용되는 등 학자들간의 합의된 기준 없이 사용되고 있다.

일반적으로 소프트웨어 개발자를 정의함에 있어서는 소프트웨어 개발자들이 수행하는 직무를 중심으로 개념을 정의하나, 우리나라 연구의 경우, 국가 또는 관련 권위있는 기관에서 분류한 직업의 관점에서 소프트웨어 개발자를 정의하고 있다(송경섭, 2009; 김재순 외, 2013).

첫째, 직무를 기반으로 소프트웨어 개발자를 정의한 Sonnentag, Niessen & Volmer(2007), Wynekoop & Walz(2000), Shiva(2010), McConnel(1998), 전호진 외(2012)의 연구에서는 요구분석과 프로그래밍(코딩) 직무가 공통적으로 포함되었으며 연구에 따라 테스트, 설계, 네트워크 관리, 유지보수, 기술영업, 마케팅 등의 직무가 추가되었다.

Sonnentag, Niessen & Volmer(2007)는 Software designer and programmer라는 용어를 사용하면서 컴퓨터 프로그램이 고객의 요구에 맞게 설계된 후, 에러없이 가동될 수 있게 하는 직무를 수행하는 사람이라고 정의했다. Wynekoop & Walz(2000)는 Software developer라는 단어를 사용하였으며 요구분석, 코딩, 테스트까지 일련의 과정을 수행하는 사람으로 정의했다. Shiva(2010), McConnel(1998)의 연구에서는 앞서 두 연구에서 한정된 요구분석, 설계, 프로그래밍, 테스트의 직무와 함께 네트워크 시스템 관리, 아키텍처, 웹페이지 관리, 소프트웨어유지보수 관리까지의 업무를 포함하여 Software Professionals이라는 용어를 사용하여 소프트웨어 개발자의 범위를 더 포괄적으로 정의했다. 이를 통해 알 수 있듯이 소프트웨어 개발자를 정의하기 위해서는 소프트웨어 개발자의 직무내용을 바탕으로 정의된다. 끝으로 전호진 외(2012)는 우리나라 연구로는 드물게 직무를 중심으로 소프트웨어 개발자의 개념을 정의하였으며, 설계, 컨설팅, 개발(PM, 아키텍처, 요구분석 및 설계, 소프트웨어 엔지니어링, 테스터), 관리, 기술영업 및 마케팅을 소프트웨어 개발자로 정의하였다.

둘째, 직업을 기반으로 소프트웨어 개발자의 개념을 제시한 김재순 외(2013), 송경섭(2009) 연구의 경우, 직업분류에서 해당 직업에 대한 직무내용을 상세하게 기술하고 있어 이를 기반으로 소프트웨어 개발자를 정의하였다. 대표적인 직업분류 자료로는 통계청에서 제시하고 있는 한국표준직업분류(KSCO), 취업 알선의 정보제공으로 작성된 고용노동부의 한

국고용직업분류(KECO)가 있으며, 소프트웨어 분야는 추가적으로 IT분야의 정부출연연구기관인 정보통신정책연구원(KISDI)의 IT 인력 분류, 산업별협의체인 정보통신산업진흥원(NIPA)의 소프트웨어 인력조사에 따른 분류 등이 있어 연구자별 여러 기준을 적용하여 소프트웨어 개발자를 정의하고 있다.

김재순 외(2013)의 연구는 통계청의 한국표준직업분류(KSCO)를 활용하여 컴퓨터시스템 설계 및 분석가, 시스템소프트웨어개발자, 응용소프트웨어 개발자, 데이터베이스 개발자, 네트워크시스템 개발자, 컴퓨터 보안 전문가, 웹 및 멀티미디어 기획자, 웹개발자를 소프트웨어 개발자로 정의하였다. 한국표준직업분류 외 우리나라에서 대표적으로 활용되고 있는 한국고용직업분류(KECO)에서는 컴퓨터시스템 설계 및 분석가, 네트워크시스템 분석가 및 개발자, 컴퓨터시스템 및 네트워크보안 전문가, 시스템소프트웨어개발자, 응용소프트웨어개발자, 웹마스터 및 웹개발자, 웹 및 멀티미디어 기획자, 데이터베이스 설계 구축 및 관리 기술자를 소프트웨어 개발자와 관련된 직업으로 분류될 수 있다. 또한, 송경섭(2009)의 연구는 2006년 산업자원부의 산업기술인력 수급동향 실태조사 보고서에 따라 컴퓨터 시스템 분석가, 시스템소프트웨어개발자(시스템소프트웨어엔지니어), 응용소프트웨어 개발자(응용소프트웨어 엔지니어), 데이터베이스 관리자, 네트워크 분석가 및 개발자, 컴퓨터보안 전문가, 웹개발자, 시스템 운영관리자, IT 컨설턴트의 직종을 포함한 개념으로 보았다.

선행연구를 통해 살펴보았듯이 소프트웨어 개발자는 직무와 직업을 기준으로 정의된다. 직무를 살펴본 결과, 개발대상과 무관하게 요구분석, 설계, 개발의 직무는 공통적으로 포함되었다. 또한, 그 외 기획과 보안 관련 직무가 소프트웨어 개발자의 직무와 밀접하게 연관되어 관련 있음을 알 수 있었다. 따라서 본 연구에서는 세부 대상에 관계없이 소프트웨어 분야의 직무 중 컨설팅, 영업, 마케팅, 운영을 제외한 설계, 개발, 관리, 기획, 보안 업무를 바탕으로 소프트웨어 개발업무를 수행하는 자로 소프트웨어 개발자로 정의한다.

〈표 II-2〉 소프트웨어 개발자의 정의

구분	소프트웨어 개발자의 정의	비고
직무	· 컴퓨터 프로그램이 고객의 요구에 맞게 설계된 후, 에러없이 가동될 수 있게 하는 직무를 수행하는 사람	· Sonnentag, Niessen & Volmer(2007) · Software designer and programmer로 지칭
	· 요구분석, 코딩, 테스트까지 일련의 과정을 수행하는 사람	· Wynekoop & Walz(2000) · Software developer로 지칭
	· 요구분석, 설계, 프로그래밍, 테스트, 네트워크 시스템 관리, 아키텍처, 웹페이지 관리, 소프트웨어유지보수 관리를 수행하는 사람	· Shiva(2010) · Software Professionals로 지칭
	· 설계, 컨설팅, 개발(PM, 아키텍처, 요구분석 및 설계, 소프트웨어 엔지니어링, 테스터), 관리, 기술영업 및 마케팅을 포함하여 소프트웨어 개발자	· 전호진 외(2012)
직업	· 컴퓨터시스템 설계 및 분석가, 시스템소프트웨어개발자, 응용소프트웨어 개발자, 데이터베이스 개발자, 네트워크시스템 개발자, 컴퓨터 보안 전문가, 웹 및 멀티미디어 기획자, 웹개발자	· 김재순 외(2013) · 한국표준직업분류(KSCO)를 활용
	· 컴퓨터 시스템 분석가, 시스템소프트웨어개발자(시스템소프트웨어 엔지니어), 응용 소프트웨어 개발자(응용소프트웨어 엔지니어), 데이터베이스 관리자, 네트워크 분석가 및 개발자, 컴퓨터보안 전문가, 웹개발자 · 일반 개발직무 종사자 외 시스템운영관리자, IT 컨설턴트	· 송경섭(2009) · 2006년 산업자원부의 산업기술인력 수급동향 실태조사 보고서

2) 소프트웨어 개발자의 직무

소프트웨어 개발자의 개념을 좀 더 명확히 알기 위해서는 소프트웨어 개발자들이 통상적으로 수행하는 직무에 대한 분석이 필요하다. 따라서 통계청에서 제시하는 한국표준직업분류(KSCO) 상에 제시된 소프트웨어 개발자 관련 직업들의 직무를 분석하고자 한다. 앞서 설명하였듯이 이 연구에서는 소프트웨어 개발자를 한국표준직업분류(KSCO) 상 컴퓨터시스템 설계 및 분석가, 시스템소프트웨어 개발자, 응용소프트웨어 개발자, DB개발자, NW개발자, Web 개발자, 컴퓨터 보안 전문가, 웹 및 멀티미디어 기획자를 소프트웨어개발자로 정의하였기에 이에 대한 직무를 살펴보고자 한다.

컴퓨터시스템 설계 및 분석가는 컴퓨터 시스템의 입력 및 출력자료의 형식, 자료처리 절차와 논리, 자료접근 방법 및 데이터베이스의 특징과 형식 등 컴퓨터시스템의 전반요소들을 구체적으로 결정 및 설계하고 분석하는 업무를 수행한다. 시스템 소프트웨어(소프트웨어) 개발

자는 컴퓨터 시스템의 자체기능 수행명령체계인 시스템소프트웨어를 연구 및 개발하고 설계하며, 이와 관련한 프로그램을 작성하는 업무를 수행한다. 응용소프트웨어 개발자는 기업이나 개인 등이 사용할 수 있는 워드프로세서, 회계 관리, 데이터베이스, 통계처리, 문서결재 프로그램 등 각종 소프트웨어를 개발하고 컴퓨터시스템의 사용 환경에 따라 소프트웨어의 환경을 변경하는 업무를 수행한다. 데이터베이스(DB) 개발자는 수집 자료의 효용성, 안정성 등을 확보하기 위하여 데이터베이스를 설계하고 개선한다. 데이터베이스(Data Base)를 구축할 업무를 파악하여 데이터 물리구조를 설계하고 크기를 산정하여 최적화 배치를 한다. 데이터베이스, 온라인 성능의 추이를 분석하고 소프트웨어를 변경하거나 운영을 통제하는 업무를 수행한다. 네트워크(NW) 개발자는 소프트웨어, 하드웨어 및 네트워크 장비에 관한 지식을 이용하여 네트워크를 개발, 기획하고 설계 및 시험 등의 업무를 수행한다. 웹(Web) 개발자는 웹 서버 구축 및 운영에 대한 기술적인 책임을 지며 웹의 신기술을 습득하고 적용하며, 시험하는 업무를 수행한다. 기본적인 프로그래밍 능력을 갖추고 있어야 하며, 일반적으로 시스템 통합 업체나 홈페이지 제작업체 등에서는 웹엔지니어가 웹마스터의 역할을 수행하기도 한다. 컴퓨터 보안 전문가는 해커의 해킹으로부터 온라인, 오프라인 상의 보안을 유지하기 위하여 필요한 보안프로그램을 개발하고, 보안 상태를 점검하며 보안을 위한 다각적인 해결책을 제시하는 자를 말한다. 웹 및 멀티미디어 기획자는 멀티미디어 관련 소프트웨어나 인터넷을 통해 공개되는 작품을 기획, 연출하는 업무를 수행한다.

따라서 세부적으로는 다루고 있는 대상에 따라 직무내용이 조금씩 상이하나 소프트웨어 개발 업무는 큰 범주에서의 직무영역은 거의 유사한 것으로 분석되었다. 따라서 소프트웨어 개발자의 직무영역은 크게 관련 상황 및 컴퓨터 환경에 대한 분석, 요구분석에 맞춘 설계 및 기획, 실질적으로 프로그램 언어를 사용하여 코딩화 하는 프로그램 구현, 구현과정 속 에러를 찾아내어 문제상황을 해결하기 위한 테스트, 고객 및 관련 유관 부서들과의 커뮤니케이션이 포함된 협의, 최종적으로 프로그램 구현 완료 후 고객에게 제출할 매뉴얼 형태의 보고서 작성, 납품 후 A/S 등의 품질관리, 각 단계별 발생할 수 있는 이슈에 대한 문제해결로 구조화되었다.

〈표 II-3〉 한국표준직업분류에 따른 소프트웨어 개발 관련 직업과 직무의 관계

직업 직무	컴퓨터시 스템 설계 및 분석가	시스템소 프트웨어 개발자	응용소프 트웨어 개발자	DB 개발자	NW 개발자	Web 개발자	컴퓨터 보안 전문가	웹 및 멀티미디어 기획자
분석	시스템분석 DB분석	시스템소프 트웨어분석	응용소프 트웨어 분석	DB 물리적 구조 분석	트래픽 분석		자료처리시 스템 검사	웹서버 분석
설계	구조설계 DB설계	시스템소프 트웨어설계	응용소프 트웨어 설계	DB 설계	네트워크 설계			
기획	계획수립			DB배치		서비스 기획		컨텐츠 기획
구현		시스템소프 트웨어 구현	시스템소프 트웨어 구현	DB 디렉토리 개발		웹서버구축	보안프로그 램 개발	
테스팅		테스팅	테스팅			웹 시험	보안상태 점검	
협의	고객과 협의 개발자와 협의			다른 담당자와 협의		고객과 소통	인사담당자와 논의	고객관리
보고서 작성	매뉴얼작성	명세서 설정	명세서 설정				방침, 절차, 결과 보고서 작성	
품질관리	품질검증 기술평가					웹 사이트 관리		
문제해결	절차개선	다른것과 연계	다른것과 연계		네트워크 장비 설치	고객불만 해결	해킹시 해결책 마련	

주: 한국표준직업분류에 제시된 직무내용을 기반으로 재구성. pp 177-187.

2. 전문성의 개념과 측정

전문성은 연구자마다 조금씩 상이하게 정의하고 있으나, 대체적으로 특정 분야에서 우수한 성과를 보이는 사람이 지니고 있는 능력이나 그 사람이 보이는 최고 수준의 결과물 또는 성과로 정의된다. 전문성 연구는 크게 전문가들의 사고, 인지, 문제해결과정 등의 인지과정으로 보는 인지심리학 관점, 전문성을 인적자본으로 보고 사회경제적인 가치 및 생산성으로 보는 사회경제학 관점, 교육, 훈련, 개인의 심리적 요소 등을 통해 점차 발달해나가는 과정으로 보는 HRD 관점으로 연구가 수행되었다. 전문성 개념에 대한 관점의 차이는 연구자가 속한 학문분야에 기인하기 때문에 이 절에서는 전문성의 일반적인 개념을 토대로 소프트웨어 개발자의 전문성 개념을 고찰하고자 한다.

가. 전문성의 개념

1) 전문성의 정의

표준국어대사전에서는 전문성(專門性)을 ‘전문적인 성질 또는 특성’라고 일컬으며, Oxford 사전에서는 전문성(expertise)을 ‘특정 분야에 대한 전문 지식’으로 정의하였다. 전문성에 대한 정의는 학자마다 견해가 다르기 때문에(Hoffman, Shadbolt, Burton & Klein, 1995) 명확한 정의를 내리는 것은 어렵지만 특정 분야 내에서 우수한 성과를 나타내는 사람들이 지닌 능력이나 행동이라는 점에서는 유사하다(윤형한, 2009).

전문성을 정의함에 있어 특정 분야 내에서 우수한 성과를 나타내는 사람들이 지닌 능력이나 행동이라는 점에서는 동일하지만 학자마다 견해가 다르기 때문에(Hoffman, Shadbolt, Burton & Klein, 1995) 강조하는 초점이 달라 다양하게 정의된다. 전문성을 우수한 사람들이 지니고 있는 지식, 기술, 문제해결 등의 능력으로 보는 관점, 전문가로 도달하기 위한 발달과정으로서의 관점, 우수한 성과 자체로 보는 결과론적 관점으로 구분되며 각 각의 관점을 설명하면 다음과 같다.

첫째, 능력으로 보는 관점은 전문성을 정의함에 있어 가장 일반적으로 정의되는 개념으로 특정 분야의 우수한 성과를 나타내는 사람이 지닌 능력으로 정의된다. 이 능력은 특정 직무 분야에서 요구되는 다차원적인 속성의 수준을 의미한다(Van der Heijen, 2002). 일반적으로 전문성을 갖춘 전문가들은 지식, 기술, 경험, 문제해결에 대한 수준이 높고(Harmon &

King, 1986; Van der Heijen, 2002; Kochevar, 1994; 오현석, 2006) 그 외 해당 직업에 대한 신념, 태도, 가치관, 직관, 창의성 등이 우수한 것으로 보고 전문성을 이루고 있는 구성요소를 능력으로 보고 정의하였다(이은화, 배소영, 조부경; 1995).

특히, 전문성을 특정 분야의 지식과 기술로 정의하면서 일부 학자들은 관련 지식을 중심으로 전문성을 개념화하였다. 전문성을 특정 분야에 대한 전문 지식 자체로 보는 대표적인 연구인 Winch(2010)는 전문성을 특정 영역에 대한 높은 수준의 실천적 지식(practical knowledge)으로 정의하고, 실천적 지식을 ‘아는 것(knowing that)’을 넘어 ‘할 수 있는 것(knowing how)’, 할 수 있다는 것은 실제 상황적 맥락 상에서 실제 판단할 수 있을 정도이며, 역량, 스킬, 능력 등의 단어를 포괄하는 개념으로 정의한다고 제시하였다. Alexander, Schallert & Hard(1991)도 이와 유사하게 전문성을 특정 전문 분야의 다차원적인 지식의 총합으로 정의했다. 전문성을 해당 분야의 전문지식으로 보고 이를 다차원적으로 분석하여 선언적 지식(declarative knowledge), 절차적 지식(procedural knowledge), 맥락적 지식(knowing when and where or under what conditions)이 통합된 개념으로 보았다.

둘째, 발달과정으로 보는 관점은 전문성을 우수한 성과를 도출하도록 해주는 인지적 과정(Chi & Koeske, 1983; Ericsson & Lehmann, 1999; 김정아, 2007; Nunn, 2008)으로 보거나 최고 수준의 전문가에 도달하고자 하는 개인의 발달적 노력, 능력(오현석, 2004; 이경화 & 김경희, 2006; 배을규 외, 2011)으로 정의하기도 하였다.

셋째, 수행 및 수행결과로 보는 관점은 전문성을 우수한 성과 그 자체의 결과로 본다. 따라서 특정 분야의 전문가들이 보이는 수행 수준(Webster사전, Ericsson & Lehman, 1996; 소프트웨어anson, 2001; 소프트웨어anson & Holton, 2009)으로 보기도 하고, 높은 수준을 나타내는 행동(Herling, 1998; 오현석 & 김정아, 2007)으로 보기도 하였다.

〈표 II-4〉 전문성에 대한 관점별 주요 전문성 정의 예시

관점	연구자 (연구년도)	정의
능력	Harmon&King(1986)	표준적인 보통수준을 넘어서는 수행결과를 보이는 사람들이 소유한 지식과 기술
	Alexander, Schallert & Hard(1991)	특정 전문 분야의 다차원적인 지식의 총합
	Kochevar(1994)	자신의 영역에서 전형적인 과제를 수행하는 과업-특수적 지식

관점	연구자 (연구년도)	정의
	이은화, 배소영, 조부경(1995)	직업에 조사하는 자들이 보여주는 직업에 대한 신념, 태도, 행동
	Van der Heijen(2002)	특정 직무 분야에 요구되는 직무분야의 지식, 메타인지, 기술요구, 사회 인지, 성장 및 유연성 등 다차원적 속성에 대한 지식을 갖춘 정도
	오현석(2006)	어떤 분야에서 일을 수행하기 위해 필요한 전문지식, 기술, 태도로 지식, 경험, 문제 해결로 구성됨
	Winch(2010)	특정 영역에 대한 높은 수준의 실천적 지식(practical knowledge)
	배범수 외 (2012)	개인이 보유하고 있는 직무에 대한 전문적 지식과 기술
발달과정	Chi&Koeske(1983)	정보를 훨씬 더 효율적으로 습득하고 조직하게 하여, 결과적으로 그 정보가 자신의 전문성 영역에 관련될 때 더 나은 수행에 도움을 주는 것
	Ericsson & Lehmann(1999)	전문가가 탁월한 수준의 성과를 지속적으로 수행할 수 있도록 해주는 인지적 인식 작동, 생리적 메커니즘
	오현석(2004)	실천적 성찰 학습을 통해서 해당 분야에서 최적의 수행수준을 달성하게 하는 능력
	이경화&김경희(2006)	특정 영역에서 발휘해야 하는 지식과 기술 및 이를 증진시키려는 자기주도적 학습능력, 상황에 대한 적응력, 목표를 성취하기 위한 노력
	김정아(2007)	특정 분야에서 효율적으로 실행과 뛰어난 결과를 가능하게 하는 지식과 태도를 지속적으로 확장, 개발시키는 발전적 과정
	Nunn(2008)	전문성은 한정된 것이 아니라 복잡한 사회요소들과 상황적, 맥락적으로 상호작용하면서 점차 발전해나가는 과정
	배을규 외(2011)	행위의 결과와 과정을 모두 포함하며 서로 다른 분야에서의 지식과 정보를 조합하는 개인의 발달적 능력
수행 또는 수행결과	Webster사전	인지적, 생리적, 지속적으로 우수한 수행을 보이는 것
	Ericsson & Lehman(1996)	특정 영역에서의 지속적으로 나타나는 탁월한 수행 수준
	Herling(1998)	특정 영역 및 관련 영역의 실행에서 최상의 효율성과 결과의 효과성을 보이는 개인의 행위들이 일관되게 나타나는 형태로 나열되는 행동
	소프트웨어anson(2001)	특수한 영역에서 할 수 있거나 기대된 성과에 대한 최적의 수준
	오현석, 김정아(2007)	특정 영역에서 지식과 기술을 가지고 높은 수준의 수행을 나타내는 행동이나 잠재력
	소프트웨어anson & Holton (2009)	특정 활동 영역 내에서 성과를 창출할 수 있거나 수행할 것으로 기대되는 숙련된 정도

다만 이때 정의된 전문성은 전문직(profession)과는 다르다. 전문직은 전문적인 지식과 기술 뿐 아니라 사회봉사, 독점권, 면허와 자격기준, 전문직 단체, 자율성, 윤리강령과 자기규율을 특징으로 하기 때문에 전문성(expertise), 전문가(expert)와는 다른 의미로 사용된다. 따라서 의사, 변호사, 교사 등은 전문직이라 칭하지만 미용사, 자동차정비사 등은 전문가라 지칭하되 전문직이라고 하지 않는다는 점에서 그 차이를 알 수 있다(윤형한, 2009; 명대정, 1999). 일부 연구에서 전문성을 번역함에 있어 expertise(오현석, 2006; 문승태, 김민배, 2007; 오현석, 김정아, 2007; 윤형한, 2009)와 professionalism(박준기, 2010; 강민지, 2012)을 혼용해서 번역하고 있지만 전문성 정의 및 구인화 내용을 살펴보면 professionalism을 사용한 경우 교사 등의 전문직 연구에서 주로 사용하였으며 이 때 해당 분야의 기여정도에 초점을 두었다는 점에서 큰 차이가 있다.

이러한 차이는 Mieg(2007)의 실증연구 결과가 뒷받침한다. Mieg(2007) 역시 expertise를 정의함에 있어 excellence와 professionalism이 혼재되어 있는 점을 연구문제로 삼고 이 두 개념이 expertise라는 용어로 포괄해서 사용될 수 있는지 실증연구를 시작하였다. 그 결과 excellence에 영향을 미치는 요인은 교육년수, 종사기간 등의 변인이었지만 professionalism에 영향을 미치는 요인은 사회적 인지, 공헌도 등의 변인으로 차이가 있었음을 알 수 있었다.

지금까지의 고찰을 통하여, 전문성의 개념을 정리해보면 다음과 같다. 전문성은 특정 영역에서 나타난다. 이 때 특정 영역이란 성과를 나타낼 수 있는 단위인 직업 또는 직무를 의미한다. 전문성은 수준을 내포한 개념으로, 특정 영역에서 우수한 성과를 보이는 사람들이 지닌 최고 수준을 의미한다. 끝으로 전문성은 우수한 성과를 보이는 전문가가 지닌 지식, 기술 등의 능력, 실제 전문가가 보인 성과, 도달하는 발달적 과정을 의미한다고 종합할 수 있다.

2) 유사용어와의 비교

전문성을 보다 명확히 정의하기 위해서는 전문성과 유사한 개념으로 활용되고 있는 역량과의 구분이 필요하다. 전문성과 관련된 유사용어와의 비교분석 결과는 다음과 같다.

표준국어대사전에서는 역량(力量)을 어떤 일을 해낼 수 있는 힘으로 정의하고(국립국어원, 2014) 급변하는 현대사회에서 요구되는 능력을 일컫는 개념으로 그 범위가 확대되고 있으며 직무와 관련된 능력으로 주로 기업, 경영학, 인적자원관리 분야에서 많이 활용되고 있다.

역량이라는 용어는 기존의 선천적으로 타고난 재능을 측정하는 지능검사나 적성검사로 한 인간의 수행과 관련된 능력을 측정하는 것에 한계를 나타내며 학문분야에 등장하기 시작했다(윤정일, 김민성, 윤순경, 박민정, 2007; McClelland, 1973). 학문분야에 역량이라는 ‘Competence’ 라는 용어를 처음 도입한 학자인 White는 ‘환경과 효과적으로 상호작용하는 능력(ability)(윤정일, 김민성, 윤순경, 박민정, 2007; White, 1959)’로 정의하면서 후천적으로 습득이 가능한 능력을 강조했다. 이와 같은 개념에 따라 역량이라는 용어는 직업교육, 훈련분야, 성인교육 분야에서 더욱 확대되었고 최근에는 기존 학교교육에서까지 역량 기반 수업이 강조되고 있는 실정이다(소정희, 2006).

이러한 역량에 대한 관심은 최근 OECD가 핵심역량이나 교육성과지표에 대한 담론을 활성화시키면서 전 세계적으로 확대되었다. OECD(2007)는 역량에 대해 지식(knowledge), 숙련(skills)의 보유를 넘어서는 능력으로 보고 역량을 ①이론과 개념뿐만 아니라 경험적으로 얻어진 비공식 암묵적 지식의 사용을 포함하는 인지적 역량(cognitive competence), ②기능적 능력(숙련 또는 노하우) 및 주어진 영역에서 일할 때 행할 수 있는 것들, ③특정한 상황에서 스스로 어떻게 행동할 것인지를 아는 것을 포함하는 개인의 능력(personal competence), 그리고 ④어떤 개인적·전문적 가치의 소유를 포함하는 윤리적 능력(ethical competence)을 의미한다고 정의하였다. 이러한 개념을 토대로 성공적인 삶을 영위하기 위해서 필요한 핵심역량을 강조하면서 기존 교육분야를 넘어 사회적 차원에서 전 생애적으로 고려해야 할 수 있다는 문제의식을 제기하였다(유현숙, 2002).

하지만, 국내에서 역량이라는 개념은 OECD가 제시한 범사회적인 분야의 능력을 일컫는 용어로 사용되고 있지 않으며 ‘해당업무에 대한 한 개인의 능력(Moore et al, 2002)’, ‘특정한 상황이나 직무에서 준거에 따른 효과적이고 우수한 수행의 원인이 되는 개인의 내적인 특성(Spencer & Spencer, 1993)’, ‘역량을 조직의 미션 및 전략을 달성하고, 바람직한 기업문화를 창출하는데 요구되는 지식(knowledge), 기술(skill), 태도(attitude)의 총체(김진모, 주대진, 2006)’, ‘역량이란 조직의 미션 및 전략 달성, 바람직한 기업문화 창출에 필요하고, 조직 내의 개인이 자신의 업무를 성공적으로 수행하기 위해 필요로 하는 지식, 기술, 태도의 총체(김진모, 주대진, 2006)’의 개념, 즉 특정 직무에서 우수한 성과를 내기 위한 능력으로 한정되어 사용되고 있다.

종합해보면, 전문성과 관련 유사용어들간의 가장 큰 차이점은 행동의 수준으로 볼 수 있다. 역량과 자격은 요구되는 대상을 수행해 낼 수 있는 최소한의 수준을 의미한다면 같은 대상에 대해서도 우수한 성과로 수행해내는 것을 전문성이라 할 수 있다.

3) 전문성에 대한 학문 분야별 관점

1950년대 인지적 관점으로 전환을 통해 인간은 행동 이전 인지적인 과정을 거친다라는 블랙박스가 해제되면서 혁명적인 전환이 일어났다. 즉 매우 뛰어난 성취, 수행, 업적을 보인 사람들의 인지적인 과정을 통해 그들의 내는 성과를 도출할 수 있을 것이라는 가설하에 많은 연구들이 수행되었다(Gardner, Howard, 2000; Simonton, 1977). 이 가운데 전문성에 대한 연구가 집중된 학문분야는 심리학 중에서 인지심리학, 사회경제학, HRD를 포함한 교육학 등이다. 이들은 근간이 되는 학문분야의 속성을 기반으로 전문성을 다른 관점으로 연구하였다.

Mieg(2007)은 전문성을 크게 개인이 가지고 있는 특출난 성격, 기질적인 능력을 의미하는 인지심리학적 관점(cognitive-based view)과 직업인으로서 두드러진 성과를 보이는 사회경제학적 관점(performance-based view)로 나뉘어 정의하였다(Reuber, 1997; Glaser & Chi, 1988, Frensch & Sternberg, 1989). 이와 함께 HRD와 교육학에서 최고 수준의 전문가로 도달해가는 과정을 보는 발달론적 관점으로 전문성을 고찰해보고자 한다.

가) 인지심리학적 관점(cognitive-based view)

전문성에 대한 연구는 심리학 분야에서 처음 시작되었다. 1900년대 초 파블로프, 스키너로 대표되는 행동주의 심리학자들은 자극(S)과 반응(R)의 관계에 초점을 두고 자극을 통한 행동의 변화에 관심을 두었다. 하지만 인간이 자극에 그대로 반응하는 것이 아니라 환경적 자극에 개인이 의미를 부여하거나 사고과정을 통해 같은 자극이라도 다르게 반응한다는 반박이 전개되면서 피아제, 비고츠키 등의 학자로 대표되는 인지심리학이 태동되기 시작한다.

인지심리학은 전문성 연구의 기반이 되는 이론이다. 인지이론은 인간의 사고과정에 초점을 두면서 외부 자극을 선택하고, 부호화하고, 지식을 형성하여 문제를 해결해 나가는지에 대한 정보처리 관점에서 연구가 확대되었다(이지현 외, 2009). 이처럼 1950년대 인지적 관점으로 전환을 통해 인간은 행동 이전 인지적인 과정을 거친다라는 블랙박스가 해제되면서 혁명적인 전환이 일어났다. 즉 매우 뛰어난 성취, 수행, 업적을 보인 사람들의 인지적인 과정을 통해 그들의 내는 성과를 도출할 수 있을 것이라는 가설하에 전문성의 연구의 시초가 되는 초기 연구들이 수행되었다(Gardner, Howard, 2000; Simonton, 1977).

이처럼 심리학적 접근에 뿌리를 둔 전문성 연구는 전문가가 지니고 있는 인지사고 과정, 개인 심리적 특성 등에 대한 연구로 발전되었다. 심리학 분야에서 전문성을 연구한 대표적인 학자 Erisscon 학파에서 정의한 전문성은 ‘전문가가 탁월한 수준의 성과를 지속적으로 수

행할 수 있도록 해주는 인지적 인식 작동, 생리적 메커니즘' 으로 보고 전문성을 지닌 전문가들의 인지적 사고능력에 초점을 두고 진행되었다.

심리학적 접근의 인지기반적 관점에서는 전문성을 제3자의 평가로 측정될 수 있는 기술적인 측면보다는 해당 객체가 보유한 인지적, 절차적인 능력으로 본다. 예를 들어, 전문가들의 의사결정 방법, 같은 경험을 하더라도 의도적인 연습(deliberate practice)을 통해 탁월한 결과 도출, 관련 지식의 통합적 재구조화 등 인지적으로 전문가들이 보통 수준의 사람들과 다르게 행동하는 부분에 대한 연구를 수행한다. 이러한 인지적 능력을 통해 신뢰할만한 우수한 성과가 도출되기 때문에 개인의 인지적인 능력은 전문성에 앞선 선행변인으로 간주될 수 있다.

인지적 측면을 강조한 전문성 개념에서는 학자에 따라 전문성의 구성요소로 지식, 기술, 문제해결능력에 대해 연구가 진행되었다. 특히 이러한 관점은 심리학자들에 의해서 연구되기 시작하였는데, 인지심리학자인 Ericsson & Lehmann(1999)는 전문성을 전문가가 탁월한 수준의 성과를 지속적으로 수행할 수 있도록 해주는 인지적 인식 작동, 생리적 메커니즘으로 보았고, Chi & Koeske(1983)도 정보를 훨씬 더 효율적으로 습득하고 조직하게 하여, 결과적으로 그 정보를 자신의 전문성에 연결시키는 과정으로 보았다. 또한, 공통적으로 탁월한 성과를 낼 수 있도록 하는 지식, 기술 등을 포괄한 보유 능력으로 전문성을 정의하였다(Harmon & King, 1986; Kochevar, 1994; 오현석, 2004; 이경화 & 김경희, 2006; 배을규 외, 2011).

나) 사회경제학적 관점(performance-based view)

전문성 연구가 확대되어 가면서 사회경제학자들은 전문가가 지니는 개인의 특성으로 전문성을 정의하기에는 한계가 있다는 반론을 제기하였다. 다시 말해, 전문성은 개인의 특성이라는 점에는 동의하나 사회와 상호맥락적 관계에서 사회구성원들에게 인정받은 가치라는 관점을 배제할 수는 없다는 것이다(Feltovich, Prietula, & Ericsson, 2007; Mieg, 2007).

실제 전문성에 대한 후속 연구들을 살펴보면, 전문성의 발달수준, 전문가들이 나타내는 성과, 사회적 기능 등에 대한 연구들로 확대되었으며, 전문가를 선정함에 있어서도 동료 또는 해당 분야 종사자들의 추천, 전문가와 비전문가의 비교 등 사회맥락적 상황 하에서의 기준이 적용되었다. 다시말해, 전문가들이 비전문가에 비해 사회적으로 지니고 있는 지위, 명성, 특허, 파워 등을 배제할 수 없다는 것이다(Mieg, 2007). 이러한 사회적 가치는 전문가가 보이는 성과에 기인하기 때문에 전문성을 개념화함에 있어 전문가 개인이 지니고 있는 특성과 함께 전문가들이 보이는 성과에 대한 관점도 함께 포함되어야 한다고 주장한다.

또한, 일부 사회학자들은 전문성을 사회구성원으로부터 인정받은 사회적 가치라는 관점에서 더 확대하여 전문성을 인적자본(human capital)의 한 형태라고 보기도 하였다(Mieg, 2007). 인적자본을 처음 제시한 Becker(1993)는 인적자본은 교육, 훈련, 복지 등의 투자를 통해 키워나갈 수 있는 개념으로 보았기 때문에, 이러한 관점에서 전문성 역시 해당 분야의 교육, 훈련을 통해 발달시켜나갈 수 있다고 주장했다(Mieg, 2007). 따라서 사회경제학적 관점에서의 전문성 연구는 전문가 개인이 지니고 있는 특성보다는 사회맥락상에서 전문성을 발달시켜나가는 과정, 영향요인 등에 대한 관심으로 확대되었다. 교육과 훈련을 통해 전문성이 발달될 수 있으며, 이렇게 높아진 전문성은 실제 생산성에 유의미한 영향을 미치므로 이러한 결과는 교육의 변화와 참여를 일으켰다.

사회경제학적 접근의 수행기반적 관점(performance-based view)에서는 전문성을 특정 분야의 직종에서 우수한 성과자로 주위에서 인정받는 전문가(expert)가 보이는 최고 수준의 결과 자체로 정의한다. 이러한 수행기반적 관점에서 정의한 전문성은 행동 또는 성과물 자체가 되기 때문에 제3자의 평가나 자가평가로 측정될 수 있다. 즉, 해당 객체가 보유한 인지적이고 절차적인 지식전문성을 우수한 성과 그 자체의 결과로 보는 관점으로는 인지적, 생리적, 지속적으로 우수한 수행을 보이는 것(Webster 사전)으로 정의하거나, Herling(1998)은 특정 영역 및 관련 영역의 실행에서 최상의 효율성과 결과의 효과성을 보이는 개인의 행위들이 일관되게 나타나는 형태로 나열되는 행동으로 보았다. 또한, Swanson & Holton(2001)은 특수한 영역에서 할 수 있거나 기대된 성과에 대한 최적의 수준으로 전문성을 정의하였다.

이러한 개념을 근거로 하여 수행기반적 관점에서 파생된 전문성은 직업 또는 직무를 중심으로 하기 때문에 자격(qualification), 직무능력표준(standards)와도 연계된다. 실제 수행기반적 관점의 전문성을 연구하기 위해 가장 많이 활용되고 있는 연구방법은 우수한 전문가를 추천받아 이들을 인터뷰하여 공통의 능력을 도출하는 주요사건분석법 등의 질적연구방법이다. 또한, 정부나 해당 분야의 권위있는 단체에서 전문가들이 모여 직무분석과 인터뷰를 통해 해당 분야의 직무능력표준을 만든 경우도 많다(Mieg, 2007).

다) HRD 관점(development-based view)

앞서 살펴본 인지심리학적 관점과 사회경제학적 관점은 개인이 보이는 능력이나 성과를 현재라는 한 시점으로 보는 것에 반해, HRD에서 바라보는 전문성은 개발을 통해 높아지는 과정(progress)으로 보았다. 이처럼 Nunn(2008)는 전문성을 정의함에 있어 전문성은 한정된 것이 아니라 복잡한 사회요소들과 상황적, 맥락적으로 상호작용하면서 점차 발전해나가는 과

정 자체를 의미한다고 정의했다.

따라서 HRD에서의 전문성 연구는 주로 전문성을 습득하는 방법에 관한 연구에 집중되었다. 예를 들면 개인이 전문가가 되는 과정, 전문가와 초보자의 학습방법의 차이 등 학습적인 측면에서 교육훈련개발, 형식학습, 무형식학습 등과 연계되어 연구가 수행되었으나 다른 분야에 비해 연구가 많이 수행되지는 못하였다. 하지만 Daley(1999) 등 기존 교육학 이론들과 전문성을 접목하여 다양한 학습방법들을 통한 전문성 발달 방법 등에 대한 연구가 시도되면서 최근들어 움직임이 활발해지고 있다.

특히, 전문성을 개발되어가는 하나의 과정으로 보았기 때문에 HRD적 관점의 전문성 연구는 전문성 발달 단계에 많은 관심을 보였다. 즉 어떠한 과정으로 전문성이 발달해 나아가며, 전문성을 개발하기 위해서는 어떤 요인들을 투입해야하는지에 대한 연구가 수행되면서, Dreyfus & Dreyfus(1977)의 스킬 숙련의 5단계를 시작으로, Benner(1986)의 간호사 임상등급, Fook 외(2000)의 직업 전문성 이론 등이 연구되었다.

이렇게 발달 단계로 접근하기 때문에 HRD에서의 전문성은 단계별 수행해야 하는 역량(competence)과 숙련정도(proficiency)의 개념을 포함하게 된다. Gilbert(1996)은 역량을 능숙한 행동(efficient behavior)으로 정의하였으며, Herling(2000)은 전문성의 하나의 하위 요소로 역량을 포함하기도 하였다. Herling & Provo(2000)는 역량은 일을 수행함에 있어 해당 범위의 일을 상황에 맞게 수행해내는 능력이라고 정의하면서, 전문성을 갖춘 전문가란 특정 분야에서 우수한 역량을 보유하고 있는 사람을 의미한다고 정의했다. Herling & Provo(2000)의 전문성 정의처럼 HRD 관점에서 전문성을 정의할 때에는 일을 수행할 수 있는 능력의 단일차원이 아니라 우수한 역량이라는 수준, 즉 숙련도(proficiency)가 역량(competence)과 함께 제시되어야 전문성을 정의할 수 있다. 끝으로, HRD의 전문성은 역량, 숙련도가 학습과 연계되어 있기 때문에 학습을 통해 이룬 학습성과(learning outcome)를 역량으로 보기도 한다.

라) 종합

학문분야에 따라 전문성을 보는 관점은 차이가 있었으며 이를 종합하면 다음과 같다.

인지심리학은 우수한 사람과 그렇지 않은 보통 사람이 보이는 차이를 알아보기 위해 전문성 개념을 처음 도입하였다. 우수한 성과를 보이는 사람들의 인지구조와 보통 수준의 사람들이 보이는 인지 구조의 차이들을 비교하여, 차이가 나는 부분이 무엇인지 알아가는 연구들을

지속해왔다.

사회경제학적 관점은 개인이 보유한 전문성의 수준이 사회경제학적으로 성과와 어떻게 연계되는지를 중심으로 연구가 진행되었다. 따라서 사회경제학적 관점에서는 전문성을 인적자본의 형태로 보고 성과에 직결되는 전문성을 높이기 위한 노력들을 탐색하는 연구들로 확대되었다.

HRD 관점은 인지심리 연구를 선행연구로 교육학 등에서 이루어졌다. 우수한 성과를 보이는 최고 수준의 능력을 갖기 위해서는 어떤 과정을 거쳐 발달해 나아가는지, 발달에 영향을 주는 여러 관련 변인들에 대한 연구가 진행되었다. 인지구조의 발달 뿐 아니라 인지구조를 포함한 전문가가 되기 위한 다양한 행동, 요인 등을 포괄하는 개념으로 전문성의 발달을 보았다.

따라서 개인이 특정 분야에 종사하기 위한 준비과정부터 해당 분야의 최고 수준의 전문가가 되기 까지 거쳐야하는 과정을 살펴보고 어떤 요인들로 인해 발달이 촉진되고 영향을 받는지를 연구하는 것은 필요할 것이다.

나. 소프트웨어 개발자의 전문성 구성요소

전문성은 앞서 설명하였듯이 영역특수적인 개념이기 때문에 전문성의 구성요소를 도출하기 위해서는 일반적으로 논의되고 있는 구성요소로 전문성을 정의하기에는 한계가 있다. 소프트웨어 분야는 전 세계적으로 국가별 중요한 분야로 인식되어 소프트웨어 관련 학문의 연구와 함께 국가별 소프트웨어 분야 종사자들의 전문성을 자격과 연계하여 체계적인 제도를 마련하고 있었다. 따라서 소프트웨어 개발자의 전문성 구성요소를 도출하기 위해 지금까지 수행된 선행연구와 국가별 관련 제도에 대해 고찰하기 위해 전문성의 일반적인 구성요소로 제시되고 있는 지식, 경험, 기술(문제해결)을 기준으로 고찰하고자 한다. 다만, 3가지 주요 구성요소 외 논의되고 있는 창의성, 직관, 태도, 가치 등의 구성요소는 기타로 구분하여 분석하고자 한다.

1) 일반적인 전문성의 구성요소

전문성을 갖춘 전문가로 인정받으려면 어떠한 요소를 갖추어야 하는지에 대해서는 아직까지 학자들마다 의견이 일치하지 않는다. 이는 전문성을 보는 관점에 의해 영향을 받게 되는데, 전문성을 정의할 때 얼마나 많이 보여주는가, 얼마나 잘 실천하는가, 얼마나 인정받는가

의 성과(performance)로 구분된다(Swanson, 2009). 전문성이 단순히 행위의 결과나 산출물로서 우수한 것을 의미하는데 그치지 않고 행위나 과정, 더 넓게는 잠재력까지 확장된 개념으로 활용되고 있어 전문성의 구성요소도 이와 같은 다차원적인 개념으로 접근해야 한다(배을규, 동미정, 이호진, 2011).

또한 비록 전문성이 영역특수성을 지니고 있다는 특성이 일반적으로 인정받고 있지만 지식이 끊임없이 생성, 소멸되고 빠르게 변화되는 현대사회에서 요구되는 전문가의 기준을 살펴본다면 영역특수성에 한정된 전문성에 대한 가치는 점차 그 힘을 잃게 될 것으로 전망된다. 따라서 하나의 영역에 대해 얼마나 알고 있는냐보다는 서로 다른 영역의 지식과 정보들을 얼마나 잘 조합하여 체계화시킬 수 있는가가 전문성으로 요구된다(Bereiter & Scardamalia, 1993; Greiner, 2009).

전문성의 구성요소로 일반적으로 받아들여지고 있는 것은 전통적 요소인 지식(knowledge), 경험(experience), 문제해결(solving-problem)이다(Herling, 1998).

첫째, 지식은 암묵지와 형식지로 크게 나뉜다. 형식지는 선언적 지식과 절차적 지식으로 나눌 수 있으며, 전문가에게 숨겨진 지식인 암묵지에는 비형식 지식, 감성 지식, 자기조절 지식이 있다(Bereiter & Scardamalia, 1993). 전문가에게 요구되는 지식유형은 연구자에 따라 다르지만, 공통적인 점은 전문가는 영역 특수적 지식을 더 많이 갖고 있으며(Glaser & Chi, 1988), 서로 영향을 주고받으며 전문성을 구성하지만 지식 그 자체로 전문성이 될 수는 없다는 점이다.

둘째, 전문가는 공통적으로 많은 경험을 가지고 있다. 그러나 경험이 많다는 것이 바로 전문성을 보장하지는 않는다. 전문가는 경험을 통한 성찰(reflection)의 방법에서 비전문가와 다르다. Schön(1983)은 전문가의 전문성은 누적된 지식이나 기술을 체계적으로 적용하는 과정에서 발견되는 것이 아니라 돌발적 사태에 대한 직관적 판단과 의사결정, 이러한 과정에서 축적된 지식, 즉 실천적 앎(knowing-in-action), 이를 통한 행위 중의 반성(reflecting-in-action)과 성찰적 실천(reflecting-in-practice)이라는 일련의 연속적 과정을 토대로 이루어진다고 보았다. 이러한 반성의 전 과정이 바로 전문적 실천가가 불확실하고 불안정한 독특한 갈등상황을 해결하는 기술의 핵심이라는 것이다.

셋째, 문제해결 요소는 지식과 함께 전문성의 가장 핵심적인 요소이다(Slatter, 1990). 문제 해결은 그 과정, 즉 문제를 인식하고 규정하며, 해결책을 찾고, 실행하고 반성하는 각 단계에서 전문가의 특징이 나타난다. 전문가와 초보자의 문제해결의 차이점은 문제의 표상(representation), 특히 질적 표상에서 그 차이점을 찾아 볼 수 있다(Sternberg, 1994). 즉,

전문가들의 표상은 개념적으로 더 풍부하며 잘 조직화되어 ‘심층적’ 지식에 의존하는 추상적 표상들을 사용지만, 초보자들은 개념적으로 엉성하고 피상적인 문제 표상들을 사용한다.

〈표 II-5〉 전문성 관련 선행연구에서 제시한 구성요소

구성요소	특성	선행연구
지식	<ul style="list-style-type: none"> · 분야 전문지식 · 인지적 지식, 비인지적 지식 · 전문성 수준이 높아질수록 점차 암묵지 증대 · 서술적, 절차적 지식 증대 	김현식(2005), Bonner(2007), Ericsson, Patel, & Kintsch(2000), Fincham et al.(2008), Glaser & Chi(1988), Gobet & Simon(2000), Kalakoski & Saariluoma(2001), Mayer(1992)
경험	<ul style="list-style-type: none"> · 다년간의 경험 · 현장에서의 실제적 경험 · 경험을 통해 암묵지 증대 	신종호 외(2007), 허정무(2008), Farrar & Trorey (2008), Goldman (2008), Lyneham, Parkinson, & Denholm(2008), Pugsley(2008)
문제해결 (기술)	<ul style="list-style-type: none"> · 기술을 의미하는 포괄적 개념 · 초보자는 구체적, 피상적 표상을 통해 표면적으로 문제해결하는 성향 · 전문가는 조직화된 문제표상을 통해 심층적 문제해결하는 성향 	손영우 & Dattel(2002), Bilalic & McLeod(2008), Brenninkmeyer & Spillane(2008), Powell & Willemain(2007), Sherin (2006), Spillane, White, & Stephan(2009), Torbeyns, Verschaffel, & Ghesquiere(2006)
창의성	<ul style="list-style-type: none"> · 급변하는 사회에 기존의 지식이나 틀에 얽매이지 않는 맥락적 유연성 · 지속적으로 전략을 수정하여 최상의 효율성 도출 · 더 다양한 관점을 통해 많은 가능성 고려 	박주용(2005), Bettman & Sujan (1987), Camerer & Johnson(1991), Chi et al. (1981), Cox(2007), Doyle(2008), Glaser(1996), Heller(2007), Larkin et al(1980), Ngara & Porath (2007), Schlesinger (2009), Schenk, Vitalari, & Davis(1998), Shanteau (1992), Simon & Simon(1978), Van der Heijden (1996)
직관	<ul style="list-style-type: none"> · 점차 자동적, 직관적 수행 · 최고전문가일수록 이성적, 심사숙고보다는 직관적으로 업무 수행 	Farrar & Trorey(2008), Lyneham, Parkinson & Denholm(2008), Salas, Rosen, & DiazGranados (2010), Sherin(2006)
기타	<ul style="list-style-type: none"> · 자질, 태도, 가치, 흥미 등 	김정아 & 오현석(2007), 성병환 & 정철영 (2008), 최지영(2008), Dimitriadis(2006)

주) 배을규, 동미적, 이호진.(2011). 전문성 연구 문헌의 비판적 고찰. HRD 연구, 13(1), 1-26.

이처럼 지식, 경험, 문제해결 요소는 공통적으로 전문성의 구성요소로 인식되어 왔지만, 이것만으로는 전문성이 충분히 설명되지 않는다는 것이 연구자들의 공통적 의견이다. 최근의 연구에서는 전문가들의 수행에서 보이는 창의성, 직관, 통찰력을 전문성의 구성요소로 포함시

키고 있다. 창의성이란 독창적이며 가치 있는 결과물을 만들어 내는 인지능력이다. 전문성의 과정을 그 분야의 구조적인 문제에서 정신적 자원을 계속적으로 재혁신하는 과정으로 보았을 때 전문성의 과정은 창의성과 그 맥락을 같이 한다고 볼 수 있다. 또한, 온기찬(1996)은 전문성과 직관의 밀접한 관련성을 언급하였는데, Benderly(1989)는 직관과 전문성은 별개가 아니라 동일한 것의 여러 현상들이라고 주장하였으며, Dreyfus & Dreyfus(1986)는 전문성의 개발 단계 중 5단계 즉, 전문가 수준에서 인지기능의 실행이 이성과 심사숙고보다는 직관적 사고에 의해 지배된다고 주장하는 등, 여러 학자에 의해서 전문성의 구성요소로서 직관이 제시되고 있다.

한편, 오현석(2004)은 전문가가 되기 위한 핵심요인으로 자아와 직종 가치의 통합화, 강렬한 목표 추구 행위, 바람직한 태도와 습관의 지속적인 형성을 제시하였다. 지금까지의 전문성 구성요소에 대한 연구들은 단순히 수행과정에서 보이는 특징이 중심이었던 것에 반해, 가치와 태도와 습관 등의 정의적 영역이 전문성 개발에 반드시 필요하다고 강조하고 있다.

이처럼 전문성은 구성요소 간 상호작용에 의한 지속적 학습의 과정으로서 끊임없는 지식 획득, 정보의 재구조화, 발전적 문제해결 노력의 상호 복합적 활동으로 나타나게 된다(Herling, 2000). 즉 전문성이 발달한다는 것은 전문성 구성 요소들의 속성이 질적으로 다른 차원으로 변화함을 의미하는 것으로 이해할 수 있다(배을규, 동미정, 이호진, 2011). 이에 더 나아가 전문성이 몇 개의 구성요소로 설명될 수 있는 것이 아니므로 구성 요소들 간의 관련된 상호작용, 네트워크의 개념으로 이해해야 한다는 주장이 제기되기도 한다(Nunn, 2008).

이러한 선행연구에서의 제시된 전문성 구성요소를 종합해보자면, 전문성은 지식, 경험, 기술(문제해결), 창의성, 직관, 가치, 태도, 흥미 등으로 구성된 다차원적인 요소로 구성되어 있음을 알 수 있었다. 다만, 전문성의 개념이 합의되지 않았음에도 불구하고 지식, 경험, 기술(문제해결)은 전문성의 정의가 상이하더라도 공통적으로 도출된 구성요소임을 알 수 있었다.

2) 소프트웨어 개발자의 전문성 구성요소

가) 선행연구에서 제시된 소프트웨어 개발자의 전문성 구성요소

소프트웨어 개발자를 대상으로 한 선행연구를 분석한 결과, 구체적인 용어를 함께 사용하지는 않았지만 전문성의 일반적인 구성요소로 도출된 여러 요소들과 유사하게 나타났다. 소프트웨어 개발자의 영역 특수성을 반영하여 지식 및 문제해결능력에서는 소프트웨어 라이프 사이클인 요구분석, 설계, 구현, 테스트, 관리의 절차에 따라 요구되는 능력들이 제시되었다.

다만, 전문성 구성요소 중 경험은 일반적인 구성요소에서는 1만 시간의 법칙 등과 연계되면서 해당 분야에 얼마만큼 종사했는지, 어떤 경험들을 했는지 등 경험에 대한 양적, 질적인 지표를 사용한 연구는 극히 드물었다. 해당 분야의 종사년수, 관련 교육 및 훈련 등은 전문성의 구성요소로 보아야하는지, 관련 변인으로 보아야하는지에 대해 합의된 바가 없어 (Germain & Tejeda, 2012) 연구자마다 보는 관점에 따라 다르게 적용한다.

일반적인 전문성 구성요소에 대한 통합적 고찰 연구를 한 배을규, 동미정 & 이호진 (2011)의 연구 결과를 틀로 삼아 선행연구에서 제시된 소프트웨어 개발자의 전문성 구성요소를 구분한 결과는 다음과 같다.

〈표 II-6〉 일반적인 전문성 구성요소로 본 소프트웨어 개발자의 전문성 관련 주요 선행연구

선행연구 구성요소	Sornentag (1995)	Sornentag (1998)	Sornentag (2006)	Wynekoop 외 (2000)	Chilton & Hardgrave (2004)	Prasad 외 (2010)
지식	· 기술적, 컴퓨터 관련 지식	· 문제이해 · 계획수립	· 요구분석 및 설계 · 프로그래밍 및 프로그램 이해 · 테스트와 디버깅		· 분석/설계 · 프로그래밍 · 관련 응용지식	· 개인의 자원 활용능력
경험			· 지식재구조화 및 시연		· 훈련	
문제해결 (기술)	· 사회적 스킬 · 고객지향성 · 업무스타일	· 전문적사고 · 문제의 시각화 · 피드백과정	· 커뮤니케이 션 및 협력	· 군리더십 · 순서화	· 관리, 정책 · 설득 · 리더십 · 조직커뮤니 케이션 · 협력 · 지시 · 스피팅	· 업무 효과성 · 결과물 완성도 · 적시성 · 비즈니스감 각
창의성	· 인지적스킬			· 창의적인 성격		
직관	· 동기					
기타		· 업무집중도		· 인내 · 이상적 자기 · 높은 지능	· 공감 · 인내 · 섬세함 · 듣기 · 인내 · 사교성	· 내외적 개인적 섬세함

Sonnentag는 전문성을 키워드로 수십년간 연구해온 심리학자이며 특히 소프트웨어 개발자를 대상으로 그들의 전문성에 대한 연구를 수행했다. 크게는 1995년, 1998년과 2006년에 소프트웨어 개발자의 전문성의 구성요소를 도출하는 연구를 수행하였으며, 그 결과는 전문성을 바라보는 관점에서부터 세부 영역까지 상이하였다.

Sonnentag(1995)의 연구는 전문가가 지닌 인지적 능력으로 전문성을 측정하였다. 이러한 관점에 따라 기술적/컴퓨터 관련 지식, 사회적 능력, 업무 스타일, 인지적 능력, 사용자 중심적 태도, 동기로 전문성의 하위영역을 구성하고, 소프트웨어 직무특수적인 경험과 지식, 문제해결 내용은 기술적/컴퓨터 관련 지식 영역에 포함시켰다. 사회적 능력에는 하위요소로 협업 능력, 커뮤니케이션 능력, 팀리더 능력을 포함하였고, 업무스타일의 하위요소로는 방법 및 시스템 지향적 스타일, 독립적 업무처리, 목표 중심의 계획 수립, 새로운 상황에 대한 빠른 대처, 팀단위 운영체계, 인지적 능력에서는 유연성, 창의성, 추상적 사고를 포함했다. 또한, 사용자 중심적 태도에는 사용자 입장에서 생각하기, 사용자와의 접점부분에서의 관련 지식 보유를 하위요소로 두었으며, 해당 직업에 대한 동기로 살펴보았다.

〈표 II-7〉 Sonnentag(1995)의 소프트웨어 전문성 요소

영역	세부 하위 요소	
기술적 · 컴퓨터 관련 지식	· 전문적인 경험 · 프로젝트 관련 지식	· 특정 직무 문제해결
사회적 스킬	· 협력 스킬 · 커뮤니케이션 스킬	· 좋은 동료애 · 팀 리더 역할
업무 스타일	· 방법 지향성 및 시스템적 처리 · 팀 오리엔테이션 · 독립성	· 목표 지향성 및 계획성 · 새로운 상황에 대한 빠른 적응력 · 도구의 활용
인지적 스킬	· 유연성 · 추상적 사고	· 창의성
고객 지향성	· 관련 분야의 지식 · 고객과의 접촉	· 고객입장 고려
동기	-	

Sonnentag(1998)의 연구에서의 문제이해영역은 당면 문제를 재해석하는 문제해결과정의 가장 필수적인 영역으로 이후 계획 및 실행단계의 기초가 될 수 있도록 해당 상황 또는 영역과 관련있는 정보를 취합하는 활동까지 포함한다. 따라서 보통의 우수 성과자는 보통 수준의

성과자에 비해 문제이해영역에서 소요되는 시간이 길 것으로 가정한다. 계획단계에서는 이후 실행에 선행되는 단계로 향후 프로젝트 목표를 수립하고, 이에 대한 스케줄링, 가능 자원 검토 및 확보 등을 결정한다. 따라서 우수 성과자는 보통 수준의 성과자에 비해 계획영역에서의 소요되는 시간이 길 것으로 가정한다. 피드백과정에서는 목표 대비 현재 상황의 달성 수준을 체크하는 과정이다. 이를 통해 새로운 계획이 추가되거나 기존 계획이 없어지기도 하지만 이러한 피드백 과정은 이후 결과에 정적인 영향을 준다. Dorner(1983)은 우수한 성과자일수록 기존 가정을 수정해나가기 보다는 피드백 과정을 지속적으로 거쳐 기존 수립한 가정이 옳은지 여부를 검토해 나가는 경향이 있다고 보고했다. 따라서 우수 성과자는 보통 수준의 성과자에 비해 더 자주, 더 오랜 기간 프로그램 디자인을 피드백 할 것으로 가정한다. 업무집중도에서는 업무 시간과 관련 활동을 수행할 때 다른 잡념이나 생각에 빠지지 않고 해당 일에 집중하는 정도를 의미한다. 문제의 시각화는 같은 상황을 본인 나름대로 그래프화 하거나, 도식화하여 문제를 살펴보고 기억하고 해결책을 찾아나가는 작업을 한다는 것을 의미한다. 이렇게 시각화시켜 문제를 분석하는 경우, 논리적으로 문제를 해결할 수 있고 해당 상황이 정리되어 전체를 살펴볼 수 있는 안목이 생기 때문이다(Anzai, 1991; Chi et al, 1982). 끝으로 전략적 사고는 우수 성과자와 보통 수준 전문가의 절차적 지식 영역에서 차이가 나는 가정 하에 복잡한 문제에서 우수 성과자가 성공 가능한 전략에 대한 이해도가 높다는 결과가 도출되어 전문성을 측정하는 하위요소로 포함되었다.

〈표 II-8〉 Sonnentag(1998)의 소프트웨어 전문성 요소

영역	세부내용
문제이해	· 당면 문제를 재해석하는 문제해결과정의 가장 필수적인 영역 · 이후 계획 및 실행단계의 기초가 될 수 있도록 해당 상황 또는 영역과 관련있는 정보를 취합
계획수립	· 실행에 선행되는 단계로 향후 프로젝트 목표를 수립하고, 이에 대한 스케줄링, 가능 자원 검토 및 확보 등을 결정
피드백과정	· 목표 대비 현재 상황의 달성 수준을 체크
업무집중도	· 업무 시간과 관련 활동을 수행할 때 다른 잡념이나 생각에 빠지지 않고 해당 일에 집중하는 정도
문제의 시각화	· 본인 나름대로 그래프화 하거나, 도식화하여 문제를 살펴보고 기억하고 해결책을 찾아나가는 작업
전략적 사고	· 절차적 지식 영역에서 차이가 나는 가정 하에 복잡한 문제 해결능력

Sonnentag(2006)의 연구는 앞서 살펴본 두 개의 연구와는 다르게 소프트웨어 개발자의 결과물을 중심으로 전문성을 측정하였다. Sonnentag(2006)는 선행연구들을 통해 경험의 기간(경력기간)이 전문성에 유의미한 정적 영향을 미친다고 보고하였지만, 반드시 경험이 많다고 하여 우수한 성과를 내는 전문가가 되는 것은 아니라고 보았다(Sonnentag, 1995; Vessey, 1986). 이에 따라 Sonnentag(2006)는 전문가를 경력기간을 기준으로 경험이 있는 사람과 없는 사람의 집단으로 구분하여 차이를 보기 보단, 성과를 기준으로 고성과자와 보통 수준의 성과자의 전문성의 차이를 보고자 했다. 따라서, 소프트웨어 개발자가 하는 업무를 세분화하여 요구분석, 프로그래밍, 테스트 및 디버깅, 관련 지식, 커뮤니케이션과 협업능력으로 구분하였다. 앞서 살펴본 것과 다르게 직무를 중심으로 전문성을 도출하였기 때문에 결과물을 측정하는 방법으로 볼 수 있다. 지식 관련된 부분에 있어서는 보통수준과 우수한 성과자 모두 비슷한 수준의 지식이 있었으나 우수 성과자의 경우, 같은 지식이라도 다른 영역과의 접목하여 새로운 지식으로 만들거나 같은 지식을 더 자세히 알고 있었다. 또한, 팀단위로 진행되는 소프트웨어 산업 특성상 우수 성과자는 다른 사람(리더, 동료, 사용자)과의 지속적인 커뮤니케이션을 통해 피드백의 기회를 높이는 경향을 나타냈다.

〈표 II-9〉 Sonnentag(2006)의 소프트웨어 전문성 관련 유경험자, 고성과자의 행동 비교

영역	유경험자 행동	고성과자 행동
요구분석 및 설계	<ul style="list-style-type: none"> · 문제이해에 많은 시간과 노력을 소요함 · 요구분석에 많은 시간을 소요함 · 작은 요소로 설계를 분해함 	<ul style="list-style-type: none"> · 문제이해에 적은 시간과 노력을 소요함 · 설계과정에서 적합한 대응책을 미리 생각해 냄
프로그래밍 및 프로그램 이해	<ul style="list-style-type: none"> · 추상적인 프로그램 목표를 선호함 · 추상적인 개념에 근거하여 프로그램 이해를 높임 	<ul style="list-style-type: none"> · 추상적인 프로그램 목표를 선호함 · 다차원적인 전략을 수립함
테스팅과 디버깅	<ul style="list-style-type: none"> · 비일관적 패턴을 테스트함 · 가정을 둔 절차에 따라 수행함 	<ul style="list-style-type: none"> · 문제해결을 위해 적극적으로 정보를 탐색함
지식 재구조화 및 시연	<ul style="list-style-type: none"> · 더 많고 의미단위로 지식을 재구조화함 	<ul style="list-style-type: none"> · 더 넓고, 더 디테일한 내용의 지식을 기반으로 함
커뮤니케이션 및 협력	없음	<ul style="list-style-type: none"> · 커뮤니케이션과 협력에 더 많은 시간을 소요함

1) 유경험자의 행동은 경험이 부족한 사람을 비교하여 도출한 행동임

2) 고성과자 행동은 보통수준의 성과자의 행동과 비교하여 도출한 행동임

Wynekoop 외 (2000)는 기존 선행연구에서 고성과자의 행동적인 측면만 강조된 것에 대한 제한점을 보완하기 위하여 고성과자의 행동적 특성과 함께 공통적인 성격적인 특질을 고려하였다. 이 연구는 그들이 이전에 연구한 Walz & Wynekoop(1997)의 연구를 보완한 연구로 고성과를 내는 개발자들의 특성에 대한 모델의 타당성을 확보하기 위해 진행되었다. 이를 위해 고성과자로 선정된 개발자들에게 3차례의 델파이를 거쳐 도구를 완성하였으며 설문을 통해 신뢰도와 타당도를 확보하는 절차를 거쳤으며, 측정가능한 소프트웨어 전문가의 요소는 다음과 같았다.

〈표 II-10〉 Wynekoop 외 (2000) 소프트웨어 전문가의 측정가능한 특성요소

특성요소	내용
창의적인 성격(creative personality)	· 일반적인 것과 다르게 생각하고 행동하고자 하는 경향
이상적 자기 (ideal self)	· 야망있고, 생산적이며 스스로에 대한 강한 자부심
군 리더십(military leadership)	· 자기에 대한 훈련이 강하고, 성실하며, 좋은 리더십 판단능력이 있음. 또한, 보수적이며, 생산적이고, 안정적이며, 사교적인(집단적인) 경향
높은 지능(high intelligence)	· 논리적, 분석적, 독립적, 자기확신적, 관습적인 경향
순서화(order)	· 정돈, 조직화, 계획을 강조하는 경향
인내(endurance)	· 무엇인가를 지속적으로 하는 경향

Chilton & Hardgrave(2004)의 연구는 IT 분야의 성과를 측정하기 위해 종사자들의 능력을 평가함으로써 성과를 추정하는 방법으로 접근하였다. 종사자들의 능력을 평가하는 관점은 크게 성과자체를 측정하는 효율성(effectiveness), 투자 대비 효과를 보는 생산성(productivity), 종사자들의 개인적인 특성, 성격, 기질 등을 평가하는 특성(trait), 끝으로 소속회사의 목표에 공헌하는 행동(behavioral)을 통해 성과를 추정할 수 있다고 보았다. 이 중 BARS(Behavioral Rating Scale)를 활용하여 행동을 통해 성과를 추정하는 방법을 선택했다. IT 성과에 영향을 미치는 스킬(skills)은 기술적 스킬(technical skills), 경영스킬(managerial skills), 대인관계스킬(people skills)의 3가지 영역으로 구분되며 총 17개의 하위요소를 도출하였다.

〈표 II-11〉 Chilton & Hardgrave(2004)의 IT 성과 측정 요소

요소	관련 스킬		
기술적 스킬	· 분석 및 설계	· 프로그래밍	· 기능 관련 응용 지식
경영 스킬	· 관리 · 정책 · 훈련	· 영업 및 설득 · 조직 커뮤니케이션 · 지시	· 리더십 · 스피킹
대인관계스킬	· 협력 · 듣기	· 공감 · 인내	· 섬세함 · 사교성

Prasad 외 (2010)의 연구는 소프트웨어 전문성을 측정하기 위해 도출한 성과(performance)를 측정하였으며, 성과는 가시적으로 측정하기 어려워 객관적 측정이 가능한 결과물(productivity)을 통해 측정하였다. Prasad 외 (2010)는 성과를 측정할 수 있는 변인으로 총 12개의 변인을 제시하였다. 적시성(timeliness), 자율성(autonomy), 효과성(effectiveness), 품질(quantity), 만족감(customer satisfaction), 수익성(profitability), 프로젝트 성공도(project success), 적정량(quantity), 책임감(responsibility), 멤버들의 협동력(team member's cooperation), 창의력/혁신성(creativity/innovation)을 꼽았다.

이 중 적시성은 몇 달 길게는 몇 년의 프로젝트 단위로 진행되는 소프트웨어산업 업무의 특성상 데드라인이 정해져 있는 경우가 많다. 따라서 고객이 요구한 내용을 원하는 시간에 결과물을 제공할 수 있어야하며, 한 프로그래머가 동시에 다양한 프로젝트에 참여하고 있으므로 시간을 적절하게 관리하여 적시에 결과물을 제공할 수 있어야 한다. 자율성은 자율적으로 의사를 반영해서 업무를 수행할 수 있을 때 성과가 높은 것이며, 전문성이 높을수록 성과가 높아 자율성의 비중이 늘어난다고 보았다. 일을 스마트하게 하면서 제시간에 결과물을 내는 효과성, 깔끔한 코드, 효율적으로 코드 수가 적으며, 버그가 적고 실패스가 있는 구조 등을 우수한 품질로 보았다. 또한, 사용자의 만족감과 경제적 수익성이 높고, 고객의 우수하고 긍정적인 피드백과 예산을 고려하여 수익성을 높게 본 것을 프로젝트 성공도가 높은 것으로 보았다. 또한, 아웃폴에 관계되지 않는 코드의 수가 최소로 코드를 짜는 것을 적정량으로 보았고, 일에 대한 책임감, 멤버들의 협동력, 고객의 요구를 이해하면서 새롭고 창의적으로 결과물을 도출하는 것을 창의력 또는 혁신성으로 보았다.

〈표 II-12〉 Prasad 외(2010)의 소프트웨어 전문성의 성과 지표

영역	설문내용
업무 효과성(Work efficiency)	<ul style="list-style-type: none"> · 독립적 사고(independent thinking) · 동시다발적 프로젝트 관리(handles many projects at a time) · 스킬 및 기술 마스터(mastery of skills and techniques) · 고객 관점(customer focused)
개인의 자원활용능력 (personal resourcefulness)	<ul style="list-style-type: none"> · 효율적인 프로그래밍 능력(efficient programming skills) · 고객에게 새로운 가치 창조(creates value to the customers) · 창의력/혁신성(creativity/innovation)
내적·외적 개인적 섬세함 (inner-and intra-personal sensitivity)	<ul style="list-style-type: none"> · 업무에 대한 책임감(responsible at work) · 팀원간의 협동력(cooperates with the team members)
결과물 완성도 (productivity orientation)	<ul style="list-style-type: none"> · 프로젝트 적기 완료(works overtime to complete the project) · 양적인 부분(quantity concerned) · 질적인 부분(quality conscious)
적시성(timeliness)	<ul style="list-style-type: none"> · 의사결정(takes decision) · 완료일 준수(meet deadlines)
비즈니스 감각 (business intelligence)	<ul style="list-style-type: none"> · 결과물 향상을 위한 새로운 아이디어 창출 (advocates new ideas to improve the product) · 수익성 고려(concern for profitability)

나) 국가별 능력표준에 제시된 소프트웨어 개발자의 전문성

앞서 설명하였듯이 소프트웨어 분야는 자격과 연계하여 별도의 제도를 마련하여 소프트웨어 분야의 인력을 양성하고, 능력을 신장시키기 위해 체계적인 프레임을 두고 있다. 이들 제도는 자격체계와 유사하게 직능수준과 직능유형을 기준으로 등급을 구분하고, 해당 수준에서 요구되는 능력을 제시하는 틀로 구성되어 있다. 이러한 제도들에서 전문성이라는 용어가 사용되지는 않았으나, Mieg(2007)가 제시하였듯이 전문성은 개인이 인식한 정도가 아니라 해당 분야의 전문가들이 합의한 표준지침(standard)에 부합되는 정도를 기준으로 평가되어야 하기 때문에 국가별 소프트웨어 개발자의 표준지침을 분석하는 것은 의의가 있다. 따라서 우리나라의 미래창조과학부에서 제시한 소프트웨어 개발자의 역량평가인 TOPCIT, NCS와 함께 일본의 ITSS, 영국의 SFIA에 대해서 고찰해보고자 한다.

국가별 소프트웨어 분야의 표준을 분석한 결과, 경험 영역과 창의력, 직관 등이 포함된 기타 영역에 대한 내용은 표준에 제시되어 있지 않았다. 그 이유는 국가별 소프트웨어 분야의 표준은 자격과 연계되어야 하기 때문에 객관적으로 측정이 가능한 지식 또는 기술영역에 한

정된 것으로 분석된다. 특히, 등급을 구분하는 요소로 경험, 즉 경력년수가 반영되기는 하나 소프트웨어 개발자가 지녀야하는 전문성의 구인요인으로는 제시되지 않았다.

일반적인 전문성 구성요소에 대한 통합적 고찰 연구를 한 배을규, 동미정 & 이호진 (2011)의 연구 결과인 지식, 경험, 기술(문제해결능력), 창의성, 직관 및 기타의 하위요소 중 국가별 소프트웨어 개발자의 직무능력표준으로 도출된 능력은 지식과 기술이었다. 앞서 살펴본 소프트웨어 개발자의 전문성 구성요소에서 논의된 정의적 영역 부분은 제외되었다는 것을 알 수 있었다. 그 이유는 국가차원에서 운영되고 있는 소프트웨어 개발자의 표준은 객관적인 측정이 가능해야 했기 때문이다(Hoffman, 2007; Spencer & Spencer, 1983). 따라서 객관적인 측정이 가능한 지식과 기술 영역으로 기준으로 국가별 직무능력표준에 제시된 소프트웨어 개발자의 전문성 구성요소를 구분한 결과는 다음과 같다.

〈표 II-13〉 국가별 소프트웨어 표준을 통해 고찰한 전문성 구성요소

구분	한국의 TOPCIT	일본의 ITSS	영국의 SFIA
지식	<ul style="list-style-type: none"> · 소프트웨어 개발능력 · 데이터베이스 구축 및 운영능력 · 네트워크와 보안 이해 및 활용능력 	<ul style="list-style-type: none"> · 설계 및 디자인 · 소프트웨어 엔지니어링 · 프로그래밍 기술 · 요구분석 	<ul style="list-style-type: none"> · 전략기획 · 개발 구현
문제해결 (기술)	<ul style="list-style-type: none"> · IT 비즈니스 이해 능력 · 테크니컬 커뮤니케이션 능력 · 프로젝트 관리 능력 	<ul style="list-style-type: none"> · 비즈니스 분석 · 문제해결 · 컨설팅 · 리더십 · 프로젝트 관리 · 협상 · 커뮤니케이션 	<ul style="list-style-type: none"> · 프로젝트 관리 및 운영 관리

TOPCIT(Test of Practical Competency in IT)은 국내 ICT 사업 종사자 및 소프트웨어 개발자의 비즈니스 이해 및 직무수행능력의 핵심이 되는 지식, 스킬, 태도를 종합적으로 평가하는 자격시험을 말한다(미래창조과학부, 정보통신기술진흥센터, 한국생산성본부, 2015). TOPCIT은 주로 ICT 산업에 진출하길 희망하는 신규인력의 능력평가를 위한 목적으로 개발되었다. TOPCIT은 소프트웨어 개발인력을 대상으로 한정된 시험이기 때문에 데이터베이스, 네트워크, 보안 및 시스템 관리를 주로 담당하는 직종에 대한 능력에는 적은 가중치를 두었다. 하지만, 소프트웨어 분야 특성상 이 모든 것을 두루 이해하고 있어야 업무가 효율적, 효과적으로 진행되기 때문에 역량을 측정하는 구성요소에는 포함하였다.

〈표 II-14〉 한국의 TOPCIT(2014)의 구성요소

능력		정의
기술영역	소프트웨어 개발 능력	소프트웨어를 분석·설계·개발·유지관리할 수 있는 능력
	데이터베이스 구축·운영 능력	데이터베이스의 개념과 구조에 대한 이해를 바탕으로 데이터베이스를 설계·운영·활용할 수 있는 능력
	네트워크와 보안 이해 및 활용 능력	네트워크와 보안의 기본개념과 관련 기술에 대한 이해를 바탕으로 소프트웨어 개발에 활용할 수 있는 능력
비즈니스 영역	IT비즈니스 이해 능력	비즈니스 환경을 이해하고 ICT를 활용하여 비즈니스 경쟁력을 강화하는 솔루션과 ICT산업을 이해할 수 있는 능력
	테크니컬 커뮤니케이션 능력	ICT산업과 솔루션에 대한 이해를 바탕으로 이해관계자와 말·글·매체로 의사소통할 수 있는 능력
	프로젝트 관리 능력	프로젝트의 목표 달성에 적합한 통합관리, 범위, 일정, 비용, 품질, 자원, 의사소통, 위험, 이해관계자, 프로젝트 평가를 관리할 수 있는 능력

주: <http://www.topcit.or.kr/?mCode=K01050000&lCode=KR>

국가직무능력표준상에서 소프트웨어 산업분야는 대분류 20.정보통신의 중분류 정보기술에 해당된다. 정보기술은 정보기술전략·계획, 정보기술개발, 정보기술운영, 정보기술관리, 정보기술영업의 5가지 소분류로 구분된다. 이 중 소프트웨어 개발자를 운영, 영업을 제외한 기획, 개발, 관리 업무를 수행하는 자로 정의하였으므로 소분류 중 1.정보기술전략·계획, 2. 정보기술개발, 4. 정보기술관리가 소프트웨어 개발자가 보이는 성과와 관련이 있다.

정보기술전략·계획은 정보기술전략, 비즈니스IT컨설팅, 비즈니스IT기획, 소프트웨어제품기획의 4가지 세분류로 구분되나, 소프트웨어 개발자의 정의에 따라 컨설팅은 제외하므로 외부컨설턴트가 조직환경에 대해 자문하는 비즈니스 IT 컨설팅은 제외한다. 소프트웨어 개발자의 가장 핵심적인 직무에 해당하는 정보기술개발은 대상에 따라 소프트웨어 아키텍처, 응용소프트웨어엔지니어링, 시스템엔지니어링, DB엔지니어링, NW엔지니어링, 보안엔지니어링, UI/UX엔지니어링의 7가지 세분류로 구분된다. NCS상에서 엔지니어링의 표현은 한국표준직업분류와 한국고용직업분류상의 개발자 또는 설계자와 동일한 의미를 가진다. 끝으로 정보기술관리는 IT프로젝트관리, IT품질보증, IT테스트의 3가지 세분류로 구분되며 이는 소프트웨어 개발자가 수행해야하는 업무 중 관리 부분에 해당된다.

일본의 ITSS(Information Technology Skill Standard)(2010)는 IT 기술분야를 직종에 따라 11개의 직종, 38개의 세부전문분야로 나눈 후 레벨에 따른 필요한 스킬을 제시하였다. ITSS가 제시한 직종은 마케팅, 세일즈, 컨설턴트, IT컨설턴트, 프로젝트 관리, IT 전문가, 애플리케이션, 소프트웨어, 고객서비스, 오퍼레이션, 교육으로 구분하였으며 그 중 소프트웨어 개발자에 해당하는 직종으로 컨설턴트, IT 아키텍처, IT 전문가, App전문가, 소프트웨어개발자, 프로젝트매니저에 대한 능력을 분석하였다. 일본의 ITSS도 우리나라의 TOPCIT, NCS와 유사하게 관련 분야의 경험, 창의력, 직관력 등에 대한 요소는 없었다. 다만, 대체적으로 지식과 관련된 영역에서는 소프트웨어 개발의 라이프 사이클을 고려하여 요구분석, 설계, 프로그래밍 등의 능력을 제시하였으며, 그 외 협상, 커뮤니케이션, 프로젝트 관리, 리더십 등의 능력을 제시하였다.

영국의 SFIA(Skill Framework for Information Age)는 영국의 IT 관련 주요 단체들이 연합하여 개발한 IT 스킬 표준 프레임워크를 말한다. SFIA는 직종에 따라 전략기획, 운영관리, 판매마케팅, 개발구현, 서비스 제공의 5가지로 구분하였다. 학자 및 국가에 따라 지칭하는 용어가 상이한 경우가 있는데 SFIA에서 이 중 이 연구에서 정의한 소프트웨어 개발자에 해당하는 전략기획, 개발구현을 중심으로 고찰하였다. 전략기획 영역에서는 비즈니스 영역과 기술영역에 대한 전략을 세우기 위해 시장을 분석하고, 위험요소 등을 관리하는 기획이 포함되어 있었으며, 개발 구현영역은 데이터분석, 기술인증, 시스템 설계, 프로그램 개발, 테스트 등 이 역시 소프트웨어 개발의 라이프사이클이 포함되어 있었다. 그 외 일본의 ITSS와 다르게 리더십, 커뮤니케이션 등의 능력들을 별도로 제시하지는 않았다.

3) 종합

소프트웨어 개발자의 전문성에 대한 선행연구와 국가별 소프트웨어 개발 표준을 분석한 결과는 다음과 같다.

첫째, 소프트웨어 개발자의 전문성 구성요소는 일반적인 전문성 구성요소를 토대로 연구자의 관점에 따라 매우 다양하게 설정되고 있다. 위와 같이 선행연구에서 제시하고 있는 소프트웨어 개발자의 전문성 구성요소들을 종합적으로 볼 때, 전문성 구성요소는 소프트웨어 개발과 관련된 지식과 기술이 필수적으로 포함되었으며(Sonnentag, 1995, 1998; , 프로젝트를 관리하는데 요구되는 관리 능력과 운영 능력 등이 포함되어 있음을 알 수 있었다.

둘째, 소프트웨어 개발자의 전문성 구성요소는 소프트웨어 개발 생명주기의 단계별 필요한

분석 및 설계, 구현, 테스트 능력과 프로젝트 관리능력으로 유형화 할 수 있었다. 일본 ITSS와 EU e-competency에서도 소프트웨어 개발자의 전문성을 구분함에 있어 대영역으로 개발물의 개발생명주기를 기준으로 하였으며, 세부 연구들에서는 Sonnentag(2006)에서 요구분석 및 설계, 프로그래밍 및 프로그래밍 이해, 테스트와 디버깅으로 구성요소를 도출하였다. 분석 및 설계는 고객의 요구사항을 분석하는 요구분석 단계와 이를 바탕으로 전반적인 프로젝트의 개요를 짜는 설계에 필요한 지식과 기술을 의미한다. 요구사항 분석과 설계는 소프트웨어 개발 생명주기의 첫 번째와 두 번째 단계로 실제 프로그래밍이 진행되기 이전의 모든 과정이라고 볼 수 있다. 따라서 이후 이어질 전 과정에 대한 경험과 이해가 있어야 분석 및 설계를 수행할 수 있으므로 일정 기간의 숙련과 경험이 쌓인 높은 수준의 개발자들에게 요구되는 능력이다(Sonnentag, 2006; 최희선 외, 2012). 이는 Sonnentag(1998)에서 제시한 문제의 이해, 계획 수립과 관련되며, Sonnentag(2006)에서 본인의 이전 연구 결과를 좀 더 세분화하여 제시한 요구분석 및 설계와 관련이 있다. 또한, Chilton & Hardgrave (2004)의 분석 및 설계와도 관련된다고 볼 수 있다. 국가별 소프트웨어 개발능력 표준 중 하나인 한국의 TOPCIT에서는 소프트웨어 능력과 데이터베이스 구축 및 운영능력의 하위로 분석 및 설계를 포함하였으며, 일본의 ITSS에서 제시한 설계 및 디자인, 요구분석도 관련된다고 볼 수 있다.

셋째, 소프트웨어 개발자의 공통능력으로 프로그래밍, 즉 구현 능력이 도출되었다(Sonnentag, 1995, 2006; Chilton & Hardgrave, 2004). 구현은 앞서 고객의 요구사항을 분석하여 설계한 결과를 토대로 실제 프로그래밍 언어를 사용하여 결과물을 만드는 일련의 과정 전체를 의미한다. 소프트웨어 개발자에게 필수적으로 요구되는 능력으로 대부분의 입직 전 교육과정에서 집중적으로 익히는 능력이기도 하다(최희선 외, 2012). Sonnentag(1996)가 제시한 프로젝트 관련 지식과도 관련이 있으며, Sonnentag(2006)에서 제시한 프로그래밍 및 프로그램 이해, Wynekoop 외 (2000)의 기술적 지식, Chilton & Hardgrave(2004)의 프로그래밍, Prasad 외 (2010)의 프로그래밍 능력 등에서와 같이 다수의 선행연구에서 소프트웨어 개발자의 전문성 구성요소로 구현능력을 포함하였다. 국가별 소프트웨어 개발능력 표준 중 하나인 한국의 TOPCIT에서는 소프트웨어 능력과 데이터베이스 구축 및 운영능력의 하위로 분석 및 설계를 포함하였으며, 일본의 ITSS에서 제시한 프로그래밍 기술, 영국 SFIA의 개발, 구현도 관련된다고 볼 수 있다. 테스트는 완성된 결과물의 최종 검수 또는 중간 개발 결과물에 대한 중간 검수를 통해 에러 또는 버그를 고치는 과정을 의미한다. 일부 선행연구에서는 프로그래밍 능력에 테스트를 포함하는 경우도 있지만, 큰 조직의 경우 테스트 직무가 별도로 있고 개발과 다른 능력을 요구하기 때문에 Sonnentag(2006)와 Chilton & Hardgrave(2004)에서는 테스트와 디버깅 능력을 별도로 제시하였다.

넷째, 프로그램 관련 지식과 기술 외 전체 프로젝트를 관리하고 운영할 수 있는 능력이 도출되었다(Sonnentag, 1995, 1998, 2006; Wynekoop 외, 2000; Chilton & Hardgrave, 2004; Prasad 외, 2010). 프로젝트 관리는 일반적인 전문성 구성요소 중 문제해결능력과 관련이 있다. 프로젝트를 이끌어어나아감에 있어 요구되는 관리 및 운영 능력을 의미하며, Sonnentag(1996)에서 제시한 문제해결능력, 협력, 커뮤니케이션, 팀리딩, 고객고려 등과 관련이 있으며, Sonnentag(2006)에서 제시한 커뮤니케이션과 협력과도 관련이 있다. Wynekoop 외 (2000)의 문제해결능력, 의사소통능력, 조직화, 리더십 등, Chilton & Hardgrave(2004)의 관리, 리더십, 조직커뮤니케이션, 협력 등, Prasad 외 (2010)의 프로젝트 관리, 고객지향성, 협동력 등과 관련된다고 볼 수 있다. 또한, 국가별 소프트웨어 개발능력 표준 중 하나인 한국의 TOPCIT에서는 IT 비즈니스 이해 능력, 테크니컬 커뮤니케이션 능력, 프로젝트 관리 능력과 관련이 있으며, 일본의 ITSS의 문제해결, 리더십, 프로젝트 관리, 협상, 커뮤니케이션, 영국 SFIA의 프로젝트 관리 및 운영 관리도 이와 관련되어 있다.

다. 전문성의 측정방법

지금까지의 전문성 측정과 관련된 선행연구를 분석한 결과, 전문성을 능력으로 보는 관점과 성과로 보는 관점의 경계가 명확히 구분되지는 않는다. 이러한 경계의 모호함은 전문성의 정의 및 구성요소가 명확하게 규정되지 않는다는 것과 관련이 있다(오현석 & 김정아, 2010). 전문성을 연구하는 학자들 간 합의된 표준화된 전문성 측정도구는 없었다(Chalykoff & Kochan, 1989' Kidwell & Benett, 1994).

그동안 선행연구에서 나타난 전문성 측정방법은 크게 4가지 정도로 구분될 수 있다. 첫째, 지금까지 습득한 교육과 훈련을 기반으로 한 경험 위주의 심층인터뷰 방법, 둘째, 직무표준(professional standards)이나 관련 분야의 자격증(licensing)을 기준으로 해당 기준에 도달했는지 확인하는 방법, 셋째, 유사한 직무를 실제 수행하게 하여 해당 결과(performance)를 직접 측정하는 방법, 끝으로 해당 분야의 종사자들 사이에서 다년간의 경험을 통해 전문가로 인정받는 사회통념적 접근(social interaction analysis)이 있다(Hoffman & Lintern, 2006).

〈표 II-15〉 전문성 측정 방법 장단점 비교

방법	측정기준	선행연구	장점	단점
심층 인터뷰	해당 분야에 대한 인터뷰를 통해 개인의 전문성 수준을 살펴보고, 발달 과정에 대해서도 확인	Sonnentag(1995) benner(1984) Dreyfus&Dreyfus(1986) 김정아(2007) 윤형한(2007) 등	· 전문가의 경험에 대한 깊고 넓은 정보 취합 가능 · 해당 분야 종사자(동료, 상사 등)의 복수 추천을 근거로 하기 때문에 전문가에 대한 객관성 확보 가능	· 개인의 영웅담으로 기술될 소지 · 소수의 경험에 대한 과잉 일반화
객관적 표준 적용	직무표준(professional standards)이나 관련 분야의 자격증(licensing)을 기준으로 해당 기준에 도달했는지 확인	홍우림(2012)	· 국가적, 국제적으로 통용된 표준을 활용하기에 가장 객관적 기준	· 도달여부에 대한 판단 근거 타당성 미흡
유사직무 수행실험	유사한 직무를 실제 수행하게 하여 해당 결과(performance)를 직접 측정	Sonnentag(1995)	· 실제 성과수준의 객관적 평가 가능	· 설계한 실험과 직무와의 신뢰도 미흡 · 대단위 표집에 부적합한 실험연구로 한정
사회통념적 접근	해당 분야의 종사자들 사이에서 다년간의 경험을 통해 전문가로 인정을 통해	Germain & Tejada(2012) Mieg(2007) Stein(1997) ITSS(2010)	· 주변인의 설문을 통해 측정하기에 자기평가에 비해 객관적 측정 가능	· 조사수가 많은 대단위 표집에 부적합하며 질적연구에 용이 · 설문의 어려움

주) Hoffman. (1998).

Sonnentag(1995), Benner(1984), Dreyfus & Dreyfus(1986), 김정아(2007), 오현석(2006), 윤형한(2007) 등은 전문성의 발달과정을 연구하기 위해 Hoffman(1998)이 제시한 측정방법 중 심층 인터뷰를 활용하였다. 전문성 연구에 있어서 가장 중요한 절차 중 하나는 전문성을 갖춘 전문가를 연구대상으로 선정하는 일이기 때문에 해당 분야에서 다수로 추천받거나, 권위를 인정받는 자격 또는 훈장, 포상 등을 수여받은 자를 전문가로 선정한 후 심층 인터뷰를 통해 이들의 전문성을 연구하는 질적연구에서 주로 활용되고 있다. 심층 인터뷰의 장점은 전문가의 일생일대에 대한 경험을 깊고 넓게 수집하여 선행연구에서 연구되지 않았던 새로운 요인들을 찾아낼 수 있고, 선정된 전문가에 대해 신뢰도가 높은 편이다. 하지만, 개인의 일대기를 영웅담으로 기술될 요지가 있으며, 과거에 대한 회상으로 인터뷰가 진행되기 때문에 객관적인 결과로 보기 어렵다. 더욱이 시간과 비용상의 문제로 인해 5명 내외의 전문가 인터뷰로 진행되기 때문에 결과에 대한 일반화가 어렵다는 단점도 함께 가지고 있다.

홍우림(2012)은 교사의 전문성을 연구하기 위해 해당 분야의 권위를 인정받은 합의된 자격 또는 표준인 방법을 활용하였다. 관련 자격 또는 표준이 해당 분야 전문가들의 합의를 통해 도출된 결과이며, 사회적으로 통용화 되어 활용되고 있다면 더욱 객관적 지표로 적합하다. 객관적 표준 적용 방법을 통한 전문성 측정은 앞서 전문성의 개념에 대한 두 가지 관점에서 설명하였듯이 사회경제적을 기반으로 한 성과기반적 관점에 뿌리를 두고 있다. 이에 따라 전문성을 실제 개인이 나타낼 수 있는 성과의 수준정도를 전문성의 수준으로 보는 것이다. 이러한 측정방법은 전문성 측정의 가장 큰 한계점인 객관성 확보라는 문제를 해결해주기에 적합한 방법이며, 이미 선정한 자격과 표준에 대한 타당도와 신뢰도가 인정받은 경우라면 더욱 적합하다. 더욱이 연구자의 관점에 따라 적용기준을 수정, 보완하기 용이하다는 장점도 있다. 하지만 대부분의 경우, 자가평정으로 전문성 측정수준을 평가하기 때문에 실제 해당 기준에 도달하였는지에 대한 타당성을 확보하기 어렵다는 한계점이 있다.

Sonnentag(1995), Chatham(1996)가 활용한 유사 직무 수행 실험 방법은 주로 심리학 분야에서 수행되는 전문성 연구에서 많이 활용되고 있다. 객관적 표준 적용 방법이 성과기반적 관점의 전문성을 측정하는 대표적인 방법이었다면 유사 직무 수행 실험 방법은 심리학을 기반으로 한 인지기반적 전문성을 측정하는 방법이라 할 수 있다. 이는 다시 말해, 전문가가 가지고 있는 인지적인 과정을 확인하기 위해 문제해결, 창의력, 메타인지 등 컴퓨터를 활용한 다양한 실험을 통해 진행된다. 유사 직무 수행 실험 방법은 객관적 표준 적용 방법이 확보하지 못한 결과에 대한 객관적 평가가 가능하다는 것이나, 설계한 실험내용과 실제 직무 성과와의 관련성에 대한 부분이 확보되지 못하고, 시간 및 비용, 대단위 표집 등에 부적합한 실험 연구의 한계점이 있다.

ermain & Tejeda(2012), Mieg(2007), Stein(1997) 등의 연구에서 활용하였듯이 사회통념적 접근 방법은 해당 분야의 직무표준이나 자격이 없는 경우, 종사자들 사이에서 다년간의 경험을 통해 암묵적으로 인정한 개인의 전문성 수준을 의미한다. 자가평정으로 인한 전문성 측정결과와 신뢰도를 확보하기 어려워 주변인의 설문을 통해 전문성을 측정한다. 따라서 개인이 지닌 전문성의 수준에 대해 객관적인 측정이 가능하나, 설문 자체의 어려움에 대한 한계가 있다. 또한, 별도의 전문성 수준에 대한 별도의 기준이 마련되어 있지 않아 응답자마다 인식하고 있는 전문성의 수준이 상이할 수도 있다.

3. 전문성 발달과 경력경로

가. 전문성 발달 과정 및 단계

전문성은 역량, 스킬과 같이 능력을 의미하는 단어이기 때문에 측정과 평가로 이어지기 위해서는 기준에 따른 명확한 정의가 필요하다. 따라서 지금까지 전문성 단계의 대표적인 연구인 Dreyfus & Dreyfus(1977)를 포함하여 전문성(expertise)을 키워드로 발달 수준, 단계 등에 대한 연구가 진행되었지만 숙련도(proficiency level), 역량수준(the level of competency), 인지과정(cognitive process) 등 능력 발달과 관련된 연구들이 연계되어 있어 이들에 대한 종합적인 고찰이 선행되어야 한다. 따라서 능력 발달과정에 대한 연구들을 종합하여 공통적인 특징을 도출한 후, 전문성 등급에 대한 방법을 고안해보고자 한다.

첫째, 전문성을 포함한 능력의 발달은 단계로 이루어져 있다. 연구자에 따라 단계(stage)(Dreyfus & Dreyfus, 1977; Benner, 1982 등), 수준(level)(박종성 외, 2010 등)이라는 용어를 사용하고 있으나, 연속적인 개념인 능력을 구분 지었다는 것은 공통적인 특징이다. 이렇게 연속적인 개념을 구분 짓는 것은 단계마다 이전, 이후 단계와 구분되는 공통적인 특성이 있다는 점과 현장에서의 활용도에 기인한 것이라고 볼 수 있다. 현장에서 능력 측정 결과를 단계로 구분하는 것은 일반적이다. 대표적으로 정규교육과정에서 성취도 수준을 미흡, 보통, 우수 등의 수준으로 구분하거나 어학능력을 구분하고, 대학수학능력시험에서도 등급을 구분하는 것이 대표적이다.

둘째, 전문성 등 능력이 발달하는 과정은 단계를 거치며 단계의 개수는 연구자별로 상이하다. 전문성 연구의 경우 가장 대표적인 Dreyfus & Dreyfus(1977) 연구의 경우 전문성을 기술습득 수준(skill acquisition)에 따라 초보자(novice), 고급입문자(advanced beginner), 능숙자(competent), 숙련자(proficient), 전문가(expert)의 5단계를 통해 전문가로 성장한다고 보았다. Benner(1982)도 간호사의 전문성을 바탕으로 임상등급을 구분함에 있어 Dreyfus & Dreyfus(1977) 단계를 반영하여 5단계로 하였으나, 우리나라에서는 우리나라 간호 상황에 맞게 4단계(장금성, 2000)으로 변형하여 사용하지만, 미국의 일부 병원에서는 같은 간호 등급도 Benner의 초기 모형인 5단계로 활용하는 등 다양하다. 또한, Fook, Ryan & Hwakins(2000)는 직업전문성 이론(theory of professional expertise)를 제시하면서 Dreyfus & Dreyfus(1977) 모형을 기반으로 하되 초보자 전 단계로서 Pre-study단계, Dreyfus의 마지막 단계인 전문가 단계에서 경험이 많은 숙련자(experienced)와 전문가(expert practice)를 구분하여 총 7단계로 전문성을 보았다. 하지만 Bloom(1985),

Alexander(2003) 등은 능력의 수준을 측정함에 있어 가장 기본적인 수준 구분인 초급-중급-고급의 3단계를 바탕으로 한 구분을 선택하였다. Bloom(1985)는 수학, 스포츠(테니스), 음악(바이올린) 등에서 뛰어난 수행을 보이는 재능인 12명의 경력발달에 관한 4년간의 종단적 연구를 통해 그들에게 나타난 공통 특징을 중심으로 이들이 잠재력을 보이는 어린 시절로부터 어른이 되어 뛰어난 재능을 보이기까지의 전 발달 과정에 관심을 가지고 재능발달을 연구한 결과, 그들은 공통적으로 입문단계(early years), 발달단계(middle years), 완성단계(later years)의 3단계를 거치는 것으로 보았다. Alexander(2003)는 영역 학습 모형(Model of Domain Learning; MDL)에서 전문성 구성 요인들의 관계와 이것이 전문성 계발 단계마다 어떻게 달라지는지를 간략히 보여주고 있다. MDL은 학문적 영역(academic domain)에서 전문성 계발의 특성을 다요인적 다단계로 보여주고 있는데, 전문성 계발의 세 요인으로 지식, 전략적 처리, 흥미를 제시하고, 전문성 계발단계로는 적응(acclimation), 유능(competence), 숙달(proficiency)의 3단계로 구분하였다.

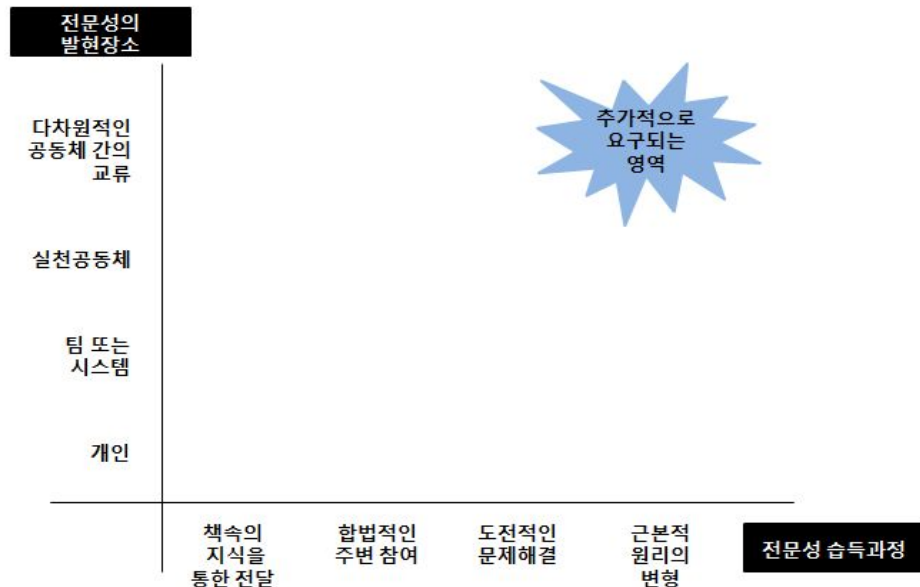
Maslow(1970) 제시한 학습의 4단계 과정은 특정 기술을 배우는 학습에 초점을 두고 설계된 것이기에 숙련의 4단계(the four stages of competence)로도 많이 활용된다. 학습의 4단계는 상황에 대한 인식과 상황을 해결할 수 있는 개인의 능력을 기준으로 무의식적 무능력 단계, 의식적 무능력 단계, 의식적 능력 단계, 무의식적 능력 단계로 구분된다. 최근에는 Maslow(1970)가 제시한 4단계를 뛰어넘는 그 다음 단계가 있다는 여러 학자들의 주장에 따라 Mata(2004)가 연구를 수행숙련에 대한 5단계를 정의하였다. 5단계는 무의식적 능력에서의 의식적 능력(Conscious competence of unconscious competence)로, 상황적 맥락에 대한 이해를 바탕으로 자동적으로 수행할 수 있는 능력을 통해 우수한 수행을 하는 것을 뛰어넘어 실제 다른 사람의 수준을 평가하거나, 최소한의 노력을 통해 최고의 성과를 낼 수 있는 방법 등을 새롭게 고안할 수 있는 정도의 수준을 의미한다. 그 외 국내에서 역량의 수준을 연구한 박종성 외(2010)의 연구에서는 Bloom의 인지발달과정을 기초로 하여 능력을 5단계로 구분하였다. 전문성을 포함한 능력을 등급으로 분류하는 개수는 일괄적으로 정해져 있기 보다는 연구자마다의 분류기준에 따라 그 개수가 상이하였다. 또한, 일반적인 개념을 넘어 분야별 특수성을 반영하여 등급이 구분되었다.

〈표 II-16〉 전문성의 발달 과정을 본 선행연구

발달 수준	인지심리학적 관점		HRD 관점				
	Maslow (1970)	Meta (2004)	Dreyfus & Dreyfus (1980)	Bloom (1985)	Alexander (2003)	Fook 외 (2000)	박종성 외 (2010)
낮음 ↑ ↓ 높음	<ul style="list-style-type: none"> · 무의식적 무능력 · 의식적 무능력 · 의식적 능력 · 무의식적 능력 	<ul style="list-style-type: none"> · 무의식적 무능력 · 의식적 무능력 · 의식적 능력 · 무의식적 능력 · 의식적 능력 	<ul style="list-style-type: none"> · 초보자 · 고급입문자 · 능숙가 · 숙련가 · 전문가 	<ul style="list-style-type: none"> · 입문 · 교육 · 완성 	<ul style="list-style-type: none"> · 적용 · 유능 · 숙달 	<ul style="list-style-type: none"> · 훈련생 · 초보자 · 고급입문자 · 능숙가 · 숙련가 · 경력자 · 전문가 	<ul style="list-style-type: none"> · 인지 · 이해 · 응용 · 지도 · 창조

셋째, 아래 [그림 II-1]은 Engerstrom(2004)가 전문성 발달에 있어 학습과정의 향후 연구 방향에 대한 도식도이다. Lave & Wenger(1991)는 실천공동체 상에서의 합법적인 주변 참여를 통해 전문성 발달이 이루어진다고 보았으며, Hutchins(1995)는 팀 또는 시스템 상에서의 합법적인 주변 참여를 통해 전문성이 습득된다고 보았다. 또한, Bereiter & Scardamalia(1993)은 개인차원이지만 도전적인 문제해결을 수행하는 과정으로 전문성이 발달될 수 있다고 제시했다.

넷째, 전문성 발달 과정은 해당 분야의 종사자들이 거치고 가는 경력경로와 연계되어 있다. Benner(1982)는 Dreyfus & Dreyfus(1977)의 기술습득을 기반으로 한 전문성 단계를 간호 분야에 적용하면서 전문성이 발달해 나아가는 과정은 간호 분야에 입문하여 최고 수준의 간호사가 되기까지 경험과 교육, 훈련을 통해 학습해 나아가는 과정으로 간호사들이 겪게 될 경력경로와 연계된다고 주장했다. 따라서 우리나라에서는 Benner의 모형을 바탕으로 4단계의 임상등급(clinical ladder)에 대한 간호역량을 제시하였다. 특히 간호분야의 임상등급은 승진, 임금 등과 연계되어 있기 때문에 그 역할이 더욱 중요하다.



[그림 II-1] 전문성 연구의 새로운 흐름도

자료: Engestrom, Y. (2004). 'The new generation of expertise', in Rainbird, H., Fuller, A., & Munro, A. Workplace learning in context, London: Routledge.

다섯째, 전문성 등 능력 발달과정은 단계별 능력의 수준이 높아지는 것과 함께 단계별 요구되는 능력 유형도 상이하다. 이종범 외(2010)은 역량이 도출된 이후 이를 평가하기 위해서는 역량의 수준과 유형을 고려해야 한다고 했다. 이처럼 역량모델의 개념적 틀을 제시함에 있어 이재경(2002)는 '역량유형 × 역량수준'을 매트릭스로 표현하여 이차원적인 수준을 제시하였다. 여기서의 역량수준은 역량유형에 대한 역량수준 정도를 의미하여, 높은 역량을 가진 사람일수록 역량수준이 높은 것을 의미한다. 다시 말해, 전문성 발달 과정을 고려함에 있어서도 단계별 요구되는 능력유형과 수준을 구분해야 할 것이다.



[그림 11-2] 역량유형 × 역량수준 매트릭스

자료: 이재경.(2002). 역량기반 교육과정 개발 방법론에 대한 고찰. 교육공학연구, 18(4), p.31.

지금까지 전문성을 포함한 능력의 등급화와 관련된 일반적인 특성을 종합한 결과, 능력의 발달은 연속적인 개념이나 단계 또는 수준으로 구분지어 설명되고 있으며, 능력을 구분하는 단계의 수는 합의된 결론없이 연구자별 별도의 기준으로 상이하다고 볼 수 있다. 다만, Dreyfus & Dreyfus(1977)과 같이 5단계가 일반적이라고 볼 수 있다. 전문성 및 능력에 대한 등급은 해당 분야의 경력경로와도 연계되기 때문에 능력의 수준이 깊어지는 것과 동시에 단계가 올라갈수록 능력유형도 변화된다는 것을 고려해야 한다.

나. 소프트웨어 개발자의 노임단가 등급

소프트웨어 산업 기술자의 등급을 분류하는 것은 소프트웨어산업법 시행령에 따른 ‘소프트웨어 기술자의 노임대가’에 제시된 등급체계를 제외하곤 국내에서 합의된 기준은 없다. 따라서 소프트웨어 기술자의 노임단가등급체계를 기준으로 국가기술자격, 공인민간자격 및 해외 자격을 분석하고, 소프트웨어 산업 관련 선행연구에서 제시한 수준을 통해 종합해보고자 한다(<표 11-17> 참고).

소프트웨어 산업 분야는 공공분야의 수주 등의 프로젝트성 사업이 많이 시행되고 있는 산업이다. 따라서 입찰방식으로 사업이 운영되고 있어 적정 가격 이하의 낙찰 등의 문제로 인해 소프트웨어 산업 종사자들의 임금이 보존되지 못해, 정부는 소프트웨어산업진흥법을 지정한 이래 법22조4항에 따라 소프트웨어 산업의 적정한 대가지급을 위해 필요한 소프트웨어기술자의 노임대가를 조사, 공표하도록 하였다(법제처, 2014). 이 후 한국소프트웨어산업협회는 통계법 제27조(통계의 공표)에 의거하여 ‘2014년 소프트웨어기술자 임금실태조사’의 결과를 공표하였으며, 이 때 소프트웨어 산업 전문가의 등급을 함께 제시하였다.

소프트웨어 산업 전문가의 등급은 자료입력원을 제외한 초급기능사에서 기술사까지 총 8단계로 구분된다. 한국소프트웨어산업협회에서 공표한 등급은 임금에 직접적으로 관여되는 기준으로 학력, 자격, 경력 세 가지 기준만을 활용하고 있다. 학력은 최종 학력으로 측정하고 있으며 자격은 국가기술자격의 수준, 경력은 총 재직년수로 측정하여 노임단가 기준에 활용하고 있다. 이 때 소프트웨어 산업에서 인정해주고 있는 자격으로는 기술자 자격의 경우 정보관리, 컴퓨터시스템응용, 기사 자격의 경우 정보처리, 전자계산기조직응용, 정보보안, 산업기사 자격의 경우, 정보처리, 사무자동화, 정보보안, 기능사 자격의 경우 정보처리 정보기기 운용 자격증이 등급 책정 시 인정받고 있다.

한국소프트웨어산업협회는 소프트웨어사업자로 신고한 업체와 협회 정회원사 등을 대상으로 이메일, 팩스, 전화조사 등을 활용하여 매년 6~7월 임금실태조사는 실시하고 있으며, 조사 결과는 매년 8월 말에 제공하여 당해도 9월부터 적용할 수 있도록 공표하고 있다. 2014년 한국소프트웨어산업협회에서 8월 제시한 노임단가 및 기술자 등급 기준은 <표 II-17>과 같다.

<표 II-17> 소프트웨어 산업 기술자의 등급 분류 기준표

구분	자격 조건
기술사	· 기술사 자격을 소지한 사람
특급기술자	· 고급기술자 자격 취득 후 3년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람
고급기술자	· 중급기술자 자격 취득 후 3년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람 · 박사학위를 가진 자로서 기사자격을 취득한 자

<계속>

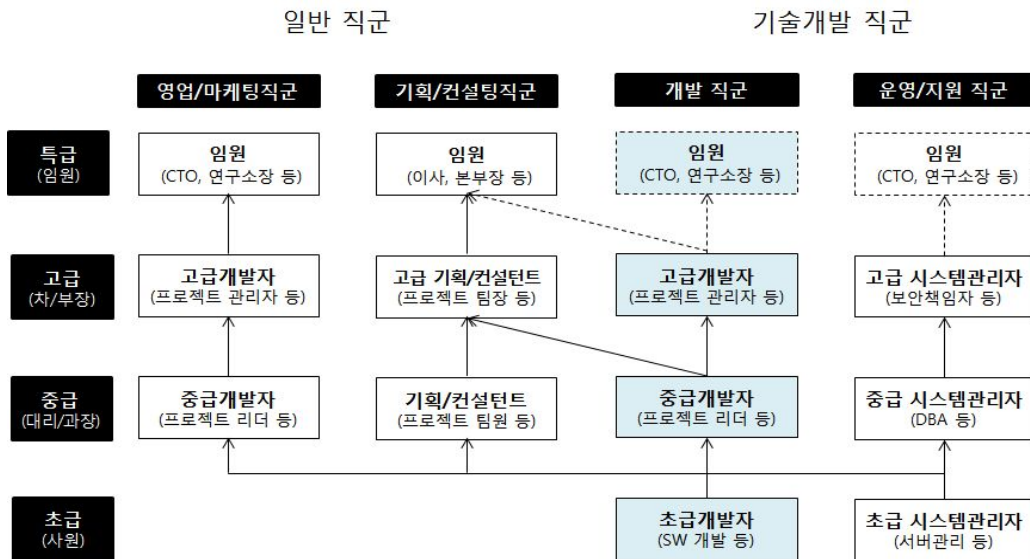
구분	자격 조건
중급기술자	<ul style="list-style-type: none"> · 기사 자격을 취득한 자로서 3년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람 · 산업기사 자격을 취득한 자로서 7년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람 · 기사자격을 취득한 자로서 석사학위 취득 후 2년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람
초급기술자	<ul style="list-style-type: none"> · 기사 자격을 취득한 자 · 산업기사 이상의 자격을 취득한 자 · 전문학사 이상의 학위를 가진 자 · 고등학교 졸업한 후 3년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람
고급기능사	<ul style="list-style-type: none"> · 산업기사의 자격을 취득한 자로서 4년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람 · 기능사의 자격을 취득한 자로서 7년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람
중급기능사	<ul style="list-style-type: none"> · 산업기사의 자격을 취득한 자 · 기능사의 자격을 취득한 자로서 3년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람
초급기능사	· 기능사의 자격을 취득한 자
자료입력원	-

노임대가를 위한 소프트웨어 기술자의 등급분류 기준을 살펴보면, 기능사와 기술사로 구분된다. 기능사는 학력에 상관없이 기사 이하의 관련 자격증을 소지하고 이거나 해당 분야의 종사기간의 자격 조건을 갖추고 있는 반면, 기술자는 전문학사 이상의 학위를 가진 자, 즉 고등학교 졸업 이상인 사람이 처음 입직할 때 해당될 수 있는 수준이다. 이 연구에서 선정한 소프트웨어 개발자에게 요구되는 자격 요건 중 학력은 학사 이상인 점을 감안하면 이 연구 대상인 소프트웨어 개발자는 소프트웨어 기술자 노임단가등급분류 기준에서는 초급 기술자 이상인 기술자 등급에 해당됨을 알 수 있다.

다. 소프트웨어 개발자의 경력경로 모형

우리나라 소프트웨어 개발자의 경력을 살펴보면 4년제 대학 졸업 후 특급단계(임원)에 도달하는데에는 평균 15년의 기간이 소요되며, Cluster 2(중급개발자) 이후 핵심 직무 기술에 있어 소프트웨어 전공 지식과 경험을 직접적으로 요구하는 기술개발 직군과 일반적인 관련 지식과 기술을 보유하고 있으나 직무수행에 핵심적으로 요구되지 않는 일반직군으로 구분되었다(최희선, 김주영, 조진환, 2012). 이는 앞서 살펴본 영국의 사례와 유사하다. 다만, 다른 나라에 비해 기술개발 직군으로 경력경로를 쌓아가는데 있어 보유 기술 수준과 부합하지 않는 임금체제, Cluster 3(고급개발자) 이후 기술개발직군에서의 임원 승진의 어려움 등 현실적인 이유로 많은 한계에 부딪히고 있다.

소프트웨어 종사인력의 경력경로를 나타낸 [그림 II-3]를 살펴보면, 초급단계에서는 개발자와 시스템관리자 2개의 직군으로 시작하다 중급단계부터 4개의 직군으로 구분되어 경력이 개발된다. 영업/마케팅 직군과 운영/지원직군은 중급단계에서부터 업무 특수성으로 인해 고유의 트랙으로 경력이 쌓이지만 개발직군과 기획/컨설팅 직군은 앞서 설명한 보유기술 수준에 부합하지 않는 임금체제, 승진 등의 문제로 인해 개발자가 기획/컨설팅 직군으로 이동됨을 알 수 있다.

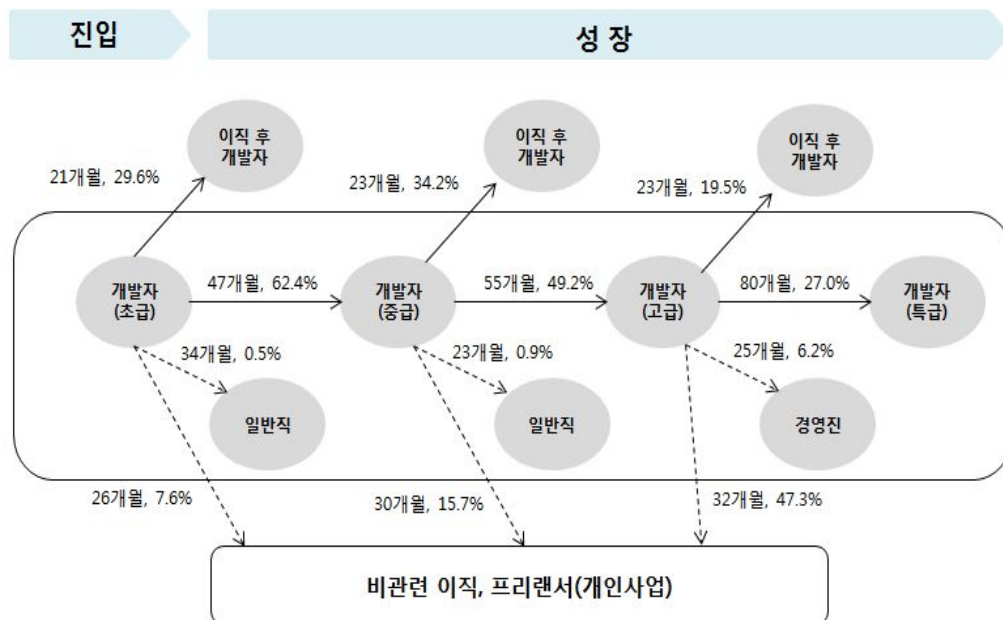


[그림 II-3] 소프트웨어 종사인력의 경력경로

자료: 정보통신산업진흥원. (2011). 소프트웨어 직무수행능력표준. p.51

기술개발 직군에 있는 소프트웨어 개발자의 경력경로를 자세히 살펴보기 위해, 최희선 외 (2012)가 수행한 연구 결과를 추가적으로 고찰하였다. 다른 국가에 비해 적은 비율이기는 하나 초급 개발자부터 특급까지 27%가 기술개발 경력을 쌓고 있는 것으로 나타났다. 중급 개발자 단계에서는 다른 회사로 이직하는 비율이 높았으나, 고급 개발자 수준에 도달하면 특급 기술자로 가는 비율은 27%에 불과하며 소프트웨어 외 분야로 이직하거나, 프리랜서 개발자로 전향하는 경우가 47.3%에 달했다. 이를 앞서 계속 제시하였듯이 고급 개발자가 임금, 승진 등의 이유로 산업에서 적합한 대우를 받고 있지 못하고 있음을 반증하는 것으로 볼 수 있다.

또한, 초급, 중급, 고급, 특급 개발자의 경력경로 단계별 요구되는 재직기간을 살펴보면 입직 이후 47개월(약 4년)이 지나면 중급 개발자로 인정되며 Cluster 2(중급개발자) 진입 후 55개월(약 4.7년), 입직 이후 8.7년이면 고급 개발자가 된다. 이후 Cluster 3(고급개발자)에서 Cluster 4(특급개발자)로 인정받는데 80개월(약 6.8년)이 요구되며, 입직 이후 15.5년이 최고 수준인 특급 개발자로 인정받는다를 것을 알 수 있다.



[그림 II-4] 소프트웨어 전문인력(기술개발직군)의 경력경로

자료: 최희선, 김주영, 조진환(2012). 산업기술인력의 경력경로에 관한 연구: 소프트웨어 산업을 중심으로 산업연구원 p.88.

이처럼 특정 분야의 전문성 등급을 분석하기 위해서는 해당 분야 종사자들의 경력경로를 고찰해야하기 때문에(강금성, 2000), 소프트웨어 개발자를 포함한 이공계 기술인력의 경력경로를 살펴보았다. 기술인력의 경력경로는 대부분 복선형(이중경로시스템, dual ladder system)의 형태를 이룬다. Bailyn & Lynch(1983)은 엔지니어의 경력경로를 설명하면서, 일정 수준에 도달한 후 승진시점에서의 경력경로를 해당 분야의 기술직으로 경력을 쌓아가는 트랙과 일반 관리업무를 수행하는 트랙으로 나누었다. Bailyn & Lynch(1983)은 90명 학사 출신의 엔지니어를 종단연구하면서 이들의 경력경로를 본 직무를 지속적으로 쌓아가는 트랙, 관리자 트랙, 다른 분야로의 이직 트랙으로 구분하였다(최희선, 김주영, 조진환, 2012). 이처럼 소프트웨어 개발자의 경력경로도 복선형이며, 프로그램 개발 직무에서 전문성을 쌓아가는 기술직 트랙이 있는 반면, 일정 수준에 도달하면 프로젝트 총괄관리를 넘어 기획, 컨설팅, 마케팅, 영업 등의 일반 관리 직군으로 진로를 변경하는 일반직 트랙으로 구분된다.

다른 국가에서도 소프트웨어 개발인력의 경력경로에 대해 유사하게 제시하고 있다. 예를 들어 영국 NOS(2009)는 IT 인력의 경력개발경로를 이러한 이중경력시스템을 반영하여 해당영역에서의 전문가형 유형(professional)과 프로젝트 관리자(manager)로의 경로를 구분하여 제시하고 있다. 반면 홍콩대학교(2006) 제시한 소프트웨어 개발자의 경력경로모형을 보면, IT 전체인력의 경력경로를 살펴본 영국의 NOS와 달리 영업, 마케팅, 기획 등의 직무를 소프트웨어 개발 직무로 정의하지 않고 구현과 관련된 직무만을 소프트웨어 개발로 정의하였다. 따라서 상위 수준에서 제시한 Manager와 Director는 우리나라 정보통신산업진흥원이나 영국의 NOS에서 언급한 일반 트랙의 관리자와 다른 의미를 가진다.

지금까지 특정 분야의 전문성 등급을 분석하기 위해서는 해당 분야 종사자들의 경력경로를 고찰해야하기 때문에(강금성, 2000), 소프트웨어 개발자의 국내외 경력경로를 살펴보았다. 소프트웨어 개발자는 경력년수, 수행직무 등을 기준으로 연구에 따라 상이한 경력단계를 이루고 있었다. 소프트웨어 산업 기술자 등급은 학력, 기간, 자격증을 기준으로 9단계로 구분하였으며, e-Competence framework, 영국의 IT 인력의 경력개발경로, 홍콩대학교의 소프트웨어 개발자의 경력경로모형은 대체적으로 5단계로 구분하였다. 역량발달, 전문성 발달, 능력발달의 개념은 직업자격체계 내의 수준에 따라 요구되는 능력별 차이점을 반영해야하기 때문에(Winterton, Le Deist & Stringfellow, 2005; Daniels, Erickson & Dalik, 2001; Winterton & Winterton, 2002b; Winterton et al, 2000) 단일직종으로 초급부터 전문가까지 같은 직무를 수행하지 않는 소프트웨어 개발자의 경우 수준별 수행하는 직무를 분석하는 것은 매우 중요하다. 따라서 소프트웨어 개발자의 경력경로와 전문성 발달 수준을 종합하면 초급~중급까지는 개발 직무를 주로 수행하였으며, 중급이상으로 갈수록 관리와 기획/설계의 직무를 주로 수행하는 것을 알 수 있었다.

라. 전문성 등급 구분 방법

특정 변인에 대해 등급을 구분하는 연구는 통계학, 경영학, 교육학 등에서 많이 활용되고 있다. 특히 통계학에 근간을 두고 신용평가등급을 하는 분야에서는 등급을 결정하는 변인에 대한 선정, 등급결정 방법들이 발달해 있다. 최근 이러한 등급 결정방법을 활용하여 경영학 분야에서는 마케팅 자료로 고객등급화모형에 대한 연구들이 많이 진행되고 있다. 반면, 경영학 또는 교육학에서 조직구성원의 역량평가나, 학생들의 성취수준평가, 자격체계 등에 활용되고 있으며 이는 교육측정 분야에서 중요한 주제로 연구되고 있다(성태제, 2011).

이처럼 수준 또는 등급을 구분하는 접근방법은 학문분야와 학자에 따라 조금씩 상이하다. 보통 신용평가등급이나 고객등급 등에 활용되는 방법은 회귀분석, 판별분석, 로짓분석, 프로빗 분석 등의 통계분석을 활용한 방법과, 역량평가, 성취수준, 자격체계 등에 활용되는 방법은 수준별 보여지는 행동지표나 학습성과(learning outcomes) 등의 준거설정(standard setting)을 활용한 방법으로 구분된다. 계량적인 통계방법을 활용한 경우에는 객관적 측정이 가능하다는 최대의 장점이 있으나, 전문성 연구에 적용함에 있어 전문성의 객관적 측정에 대한 학자 간 합의된 방법이 없고, 능력을 객관화한다는 점에서 상당히 제한적이므로 양적 접근 방법을 적용하는데 한계가 있다. 자격, 역량체계 등의 능력을 등급화하는 연구 및 실제 제도에서도 해당 분야 전문가들의 합의된 준거설정(standard setting)을 적용하고 있다(EALTA, 2000). 준거설정방법의 종류는 수십가지가 될 만큼 다양하지만, 패널을 이용하여 분류하는 방법은 신뢰성과 타당도를 확보하는 부분에서 미흡하다. 군집분석을 활용하여 능력을 등급화 하는 방법의 경우, 객관성이 확보됨과 동시에 등급을 구분하는 분류율로 다른 방법에 비해 매우 우수하다고 평가되고 있어(EALTA, 2000), 특히, 유럽 국가들에서 제2외국어의 능력을 등급화 하는 방법으로 많이 활용되고 있다.

준거설정 방법은 기본적으로 개인의 평가 결과에 부여된 점수를 토대로 해당 분야의 전문가 패널들이 등급을 구분하는 방법으로 시작되었으나 패널에 대한 신뢰도 문제가 제기되면서 최근에는 통합적인 방법들이 활용되고 있다(Glass, 1978; Shepard, 1979)

Sireci(2010)는 수학시험점수를 가지고 군집분석을 활용하여 집단을 구분하였다. Sireci(2010)는 교육학에서 학습결과를 구분하는 방법으로 준거설정의 다양한 방법들이 활용되고 있으나, 총점만을 기준으로 패널들이 집단을 구분하는 것은 많은 한계를 가진다고 지적하였다. 예를 들어, 수학에 대한 개인의 학습수준은 부분별로 상이할 수 있으며, 실제 평가하는 영역도 다르다. 하지만, 개별 영역을 고려하여 등급을 구분하는 것이 아니라 대부분 패널들이 개인의 수학시험 총점으로 수학 등급을 구분한다는 의미였다.

이러한 문제를 해결하기 위해 Sireci(2010)는 군집분석을 통해 등급을 구분하였다. 우선 수학에 대한 하위 영역에 대한 수준을 토대로 유사한 유형으로 묶고 그 후 그룹을 순서화시키면 기존 준거설정방법을 통해 도출하고자 하는 결과와 유사한 결과를 얻을 수 있다고 주장했다. 군집분석은 중심으로 불리는 평균벡터를 중심으로 각 표본들의 거리에 따라 집단이 분류되는 원리이므로 같은 속성을 가진 집단으로 분류하는데 적합하다고 보았다.

군집분석을 통해 개인 능력의 수준을 구분하는 것은 군집분석이 본래 지니고 있는 단점에서 자유로울 수 없기에 이를 극복할 수 있는 다른 방법을 추가해야한다. 군집분석을 활용하는 장단점을 살펴보면 다음과 같다. 앞서 설명하였듯이 군집분석은 총점을 활용하지 않으므로 평가에 포함된 다양한 요소와 수준을 반영해서 유사한 군집으로 구분할 수 있다는 대표적인 장점이 있다. 특히 소프트웨어 개발자의 경우 전문성 구성요소로 논의된 분석 및 설계, 구현, 테스트 및 프로젝트 관리가 전문성 발달 수준에 동등하게 영향을 미치는 것이 아니라 경험이 쌓이고 직급이 올라갈수록 분석 및 설계, 프로젝트 관리 요소들의 중요도가 더 높아지고, 초기 경력자의 경우에는 구현과 테스트 능력이 더 요구된다는 선행연구의 결과를 반영해야 하기 때문에 총점보다는 군집분석을 활용하는 것이 더 적합하다(최희선 외, 2012). 하지만 이러한 장점은 큰 오류와 연결될 수 있다. 군집분석은 최저 점수부터 최고 수준까지 일렬로 나열시켜 등급을 구분하는 것이 아닌 요소들의 수준을 반영해서 유사한 집단들을 묶는 것이기 때문에 일부 표본들은 기본적인 숙련수준과 동떨어지게 분류될 위험이 발생할 수 있다. 따라서 활용하고 있는 구성요소가 아닌 제 3의 구성요소, 랜덤한 방법으로 2~3개 그룹으로 데이터를 추출한 후 통계적으로 동일하게 구분되는지 검증할 필요가 있다(Sireci, 2010).

4. 전문성과 교육훈련, 자격 및 직무경력 관계

전문성이 발달해 나아가는 과정은 학습과 연결된다. 전문성 발달과정에 대한 연구와 함께 교육학과 심리학에서는 발달을 위한 학습유형, 개인의 특성, 개인의 의지, 조직환경적인 지원 등 다양한 요인들과의 관계에 대한 연구를 진행해왔다. 따라서 오현석(2007)의 연구와 같이 학습에 기반한 전문성 발달과정을 분석하기 위해 Gradner의 개인, 영역, 환경의 체계적 접근법(IDF)에 따라 선행연구를 고찰해보고자 한다.

첫째, 개인(I) 부분은 개인이 해당 학습에 대해 지니는 흥미, 감정, 의지 등을 의미한다. 개인영역에 해당하는 개인의 지능, 관심, 선호도 등은 전통적으로 학습에 영향을 미치는 변인으로 논의되어 왔다. 최근 전문성에 대해 세계적으로 연구하고 있는 Ericsson은 전문성의 발달과정에서 중요한 키워드는 ‘의미있는 학습(deliberate practice)’이라 강조하고 있다(Ericsson, 2007). 1만 시간의 법칙이라는 단어가 등장하면서 특정 분야에서 10년 종사하면 전문가가 될 수 있다라는 주장이 제기되었지만, 모두가 10년을 투자해도 전문성에 차이가 난다고 보았다. 그 이유는 개인이 10년이라는 시간 동안 겪게될 많은 경험에 대해 얼마나 의지를 가지고, 다시 성찰하고, 다시 적용하여 나만의 방식을 만들어가고 극복해 나아가는지가 얼마나 해당 분야에 종사했는지에 대한 기간보다 전문성 발달에 결정적인 영향을 미친다는 것이다. 따라서 개인영역에서는 개인이 지니고 있는 기본적인 학습 자질과 함께 개인의 의지, 학습태도 등이 중요하게 논의된다.

둘째, 영역(D) 부분은 개인이 경험하는 활동을 의미하며 대체로 어떤 형태의 학습을 경험하는지에 대한 연구가 수행되고 있다. 전통적인 정규교육을 통한 학습과 함께 조직 내외에서 진행되는 다수의 교육훈련인 형식학습은 학습유형의 태도적인 연구주제이다. 이와 함께 최근에는 무형식학습(홍우림, 2012; 윤형한, 2009; 이지혜, 문용린, 2011), 경험학습(한상만, 이희수, 2014), 일터학습(이재실, 2011; 김재순, 강예지, 장원섭, 2013)을 통한 학습의 중요성이 부각되면서 전문성 발달을 위한 중요한 학습형태로 연구되고 있다.

셋째, 환경적인 측면은 전문성 발달을 촉진시켜주는 멘토의 중요성에 대한 연구가 다수를 이루었으며 조직의 지원, 부모의 지원, 또래, 교사 등의 환경적인 지원이 전문성 발달을 촉진한다고 보았다.

따라서, 지금까지 전문성 발달에 영향을 미치는 변인으로 연구된 것들은 개인차원의 영향으로 자기주도학습, 성찰, 결정경험, 자기효능감, 목표수립, 목표추구행동, 긍정적 태도, 자신감, 끈기, 즐거움, 열정, 리더십, 정서조절, 몰입, 가치 등과의 관계를 보는 연구(이종학,

2014; 고경미, 이선경, 심성경, 2014; 황주연, 2011; 2008; 오현석 외, 2009; 김정아, 오현석, 2007)외, 결정적인 경험, 멘토 보유 여부 또는 멘토와의 관계 등(성은모, 배형준, 강용관, 오현석, 2012; 김정아, 오현석, 2007) 개인을 둘러싼 환경적인 변인에 대한 연구가 주를 이루어왔다. 또한, 개인이 소속된 조직의 특성인 조직풍토, 전문성 지원 특성, 임파워먼트, 자율적인 분위기 등에 대한 조직 수준의 변인과의 관계연구도 수행되었다(성병환, 정철영, 2008; 이경례, 문혁준, 2013). 이들의 연구는 전문성 발달에 영향을 미치는 요인들을 구명하는 데에는 기여하였지만, 다수의 연구가 질적연구로 진행되어 전문성의 수준에 직접적인 영향이 있는지, 이에 대한 실증적인 탐색이 미흡하였다.

전문성 발달을 위한 학습을 고찰한 결과, 발달 과정은 크게 개인/영역/환경의 영향을 받아 이루어지며, 전문성 발달을 위해 개인의 잠재적 능력도 중요하지만 전문성은 영역특수적이기 때문에 교육훈련, 경험 등 실질적인 연습이 더 중요한 변인이 된다(Winterton, Le Deist, Stringfellow, 2005; Ericsson, Krampe and Tesch-Romer, 1993; Ericsson and Smith, 1991). 더욱이 일반적으로 전문성의 수준을 직접 측정하는데 많은 한계가 있기 때문에, 교육훈련, 자격 및 직무경력을 통한 대리지표를 통해 그 사람의 전문성 수준을 측정한다. 교육훈련, 자격 및 직무경력은 Engerstrom(2004)이 제시한 전문성 습득과정을 포괄할 수 있는 변인이다. 따라서 이 연구에서 선정한 3가지 변인과 전문성의 관계를 고찰해보고자 한다.

가. 교육훈련

1) 개념 및 전문성과의 관계

교육훈련은 일반적으로 교육 또는 훈련에 참여한 경험을 의미하며 학교교육, 사설교육기관에서의 교육, 사내교육, 사내연수, 기술교육 등 직무경험을 제외한 형식교육, 비형식교육, 무형식학습을 통해 이루어진다고 본다. 또한, 평생학습시대가 도래함에 따라 특정 시점의 교육이 아닌 전 생애에 걸쳐 이루어지는 평생학습(life learning)의 관점에서 정의되기도 한다(변정현, 2011).

교육훈련을 측정하는 합의된 방법은 없으나 현재 수준의 학력수준, 사설교육훈련 또는 사내교육의 참여횟수 및 시간 등으로 교육훈련수준을 측정하는 것이 일반적이다.

학력에 대한 선행연구를 보면, 김난영, 조원혁(2012)은 감사인력을 대상으로 한 전문성 연구에서 그들이 보유한 교육훈련의 수준을 학력을 대변하는 학위로 통칭하여 보았으며, 오

석영(2011)의 연구에서는 그동안 전문성 연구에 있어서 가장 많이 활용되고 있는 정규교육년수 외 학습에 대한 양적 참여와 개인의 인식에 대한 질적 측면을 함께 고려하였다. 양적 참여로는 학습유형, 참여기간을 보았으며 학습에 대한 인식으로는 교육에 대한 개인의 관심도, 직무연관성, 현장 적용성 외 학습에 참여한 요인, 학습효과에 대한 인식을 물었다. 인적자본 관점에서는 교육훈련을 평균 교육년수(Barro & Lee, 1993, 1996, 2001; Rauch, 1993; 장수명, 이번송, 2001; Simon & Nardinelli, 1996, 2002; Moretti, 2003; Shapiro, 2005; Ciccone & Peri, 2006; Rosenthal & Strange, 2008), 대졸 비율(Moretti, 2004; Acs & Armington, 2004; Liu, 2007; Abel & Gabe, 2008), 교육투자액(심재희, 2003; 김종구, 2007) 등으로 측정하기도 한다.

훈련에 대한 선행연구를 살펴보면, 연수 및 훈련 변인은 교육훈련 프로그램 성과를 보기 위한 다수의 연구에서 전문성, 역량 또는 능력 향상 등의 개념과 연계하여 수행되고 있다(마상진, 2005; 양지연, 이병준, 2007; 유현주, 2009; 김광현, 2012) 또한, 교육년수에 대한 임금수익률을 분석한 Mulligan & Sala-i-Martin(1995), Bartel(1995)의 연구들에서 학교 정규교육에 의해 설명되지 않았던 임금 상승의 원인이 직장에서의 참여한 훈련에 의한 것임이 밝혀지면서 연수 및 훈련 변인은 정규 학교교육과 함께 전문성 향상에 중요 변인으로 주목받았다. 더욱이 직무의 종류가 다양해지면서 일반 정규교육에서는 다양한 세부 직무의 능력을 갖춘 인력배출에 어려움을 느꼈다. 따라서 직장내 학교의 미스매치가 문제시됨에 따라 입직 후 기업에서의 연수 및 훈련은 이를 해결하기 위한 해결책으로 활용되었다(장혜원, 2014). 또한, 현장에서 활용되는 제도인 건설분야의 ‘건설기술자의 등급 및 경력인정 등의 기준’에서도 건설기술자의 등급 및 경력인정 등을 기준으로 역량지수를 산정함에 있어 자격, 교육훈련, 경력 외 관련 학교교육을 추가 지표로 활용하고 있다(「건설기술 진흥법 시행규칙」 제18조제10항).

연수 및 훈련 변인의 특성을 살펴보면, 단순히 연수 및 훈련에 참여한 정도도 전문성 향상에 영향을 미치지만 개인이 실제 교육훈련 내용을 통해 학습했다고 인식한 정도가 더 중요한 관점이라는 보고 있다. 이는 대개의 학습이론에서 제시하고 있는 개인의 인식여부와 같은 맥락이라고 할 수 있다. 전문성 초기 연구에서는 교육훈련 변인이 직무성과(job performance)에 그다지 중요한 변인으로 도출되지는 않았다(Hunter & Hunter, 1984). 하지만, Schmidt et al.(1979)는 교육훈련이 직무에서 요구되는 지식, 기술, 태도(KSA)와 .4의 높은 상관이 있으며, KSA는 직무성과와 .5의 높은 상관이 있다는 실증연구결과를 발표했다.

소프트웨어 산업 이외에도 일반적으로 교육훈련변인은 현장에서의 신규 인재를 채용하는 기본적인 기준으로 활용되고 있으며, 연구적인 측면에서도 전문성에 영향을 미치는 중요한

변인으로 연구되고 있다. 이미 다수의 연구에서 교육훈련과 전문성의 관계가 정적인 관계로 보고되고 있으며(김진실, 2006; 신희이, 2011; Uppal, Mishra & Vohra, 2014) 또한, 교육훈련이 전문성 및 성과에 영향을 미친다는 결과로 인해 Dokko, Wilk, & Rothbard(2009)의 연구에서는 이는 교육훈련이 성과를 예측하는 선행변인이 되는 일반적 능력이라는 결과를 근거로 하여(Schmidt, 2002) 고등교육기관에서의 교육기관을 측정하여 이를 통제하기도 하였다. 즉, 근로자의 숙련변화와 학력, 근속, 직급간의 상관관계를 추정해 보았다. 숙련변화는 학력보다는 근속과 더 높은 상관관계($-0.00 < 0.235$)를 보이며, 입직 시 숙련수준(-0.44)과 높은 상관관계 하에 있다(김미란, 김민경, 2008)

Mieg(2005)와 신원부(2013)에서도 공무원의 전문성 영향요인에 대한 연구에서 하는 전공과의 관련성을 5점 리커트 척도로 물었으며, 전문성에 .190의 유의미한 결과가 확인되었다. 윤형한(2008)의 연구 결과, 학사수준과 석사이상의 학력 간 직업상담사 전문성의 차이를 비교한 결과, 학력 간 전문성에 유의미한 결과가 확인되었다. 이는 학력과 관련하여 Bredekamp(1996)와 이연주(2004)는 보육교사를 대상으로 한 연구에서 학력수준이 보육교사의 전문성을 예언하고 높은 학력이 전문성 발달에 대한 요구를 높인다고 하였는데, 직업상담자의 전문성도 이러한 맥락에서 학력은 전문성을 제고하려는 요구와 높아가는 노력으로 해석할 수 있다. 윤형한(2008)의 연구 결과 전공 관련 여부 결과 상담 관련 전공자가 비전공자에 비해 전문성이 유의하게 높았다($p < .05$).

2) 측정방법

교육훈련에 대한 측정방법을 살펴보면 일반적으로 많이 활용하는 최종 학위(Chi, Claser & Farr, 1988; 윤형한, 2008; 김난영 & 조원혁, 2012), 교육년수(박지윤, 2010), 훈련시간(Macduffiem & Kochan, 1995), 전공(윤형한, 2008; Mieg, 2007), 훈련참여 횟수(윤형한, 2008), 훈련참여시간 등 다양한 요인들이 고려되고 있다.

먼저 가장 많이 활용되고 있는 최종 학력 방법은 교육훈련의 대표적인 측정방법으로 측정하기 용이하고 학교급을 서열로 구분할 수는 있으나, 전공, 비정규교육 등에 대한 요소가 고려되지 않고 개인의 교육훈련 이력을 자세히 반영하지 못한다는 한계를 갖는다. 하지만 활용의 용이성과 함께 현장에서 대중적으로 인식되고 있다는 장점으로 인해 소프트웨어 기술자의 노임단가 등급을 결정하는 요인으로 활용되고 있다.

이러한 한계를 극복하기 위해 학력이 아닌 교육년수(박지윤, 2010)나 교육시간(Macduffiem & Kochan, 1995)을 활용하여 측정하기도 한다. 학력과 달리 개인의 전반적인 교육훈련 이력을 등간척도화하여 다양한 통계방법을 활용할 수 있으나 학교급간의 수준을 반영하지 못한다는 큰 한계를 갖는다. 박지윤(2010)의 연구에서는 정규교육기간을 년도(year)로 측정하였으나, 교육훈련형태, 직무와의 연관성을 고려하여 살펴본 것은 의미가 있으나, 이를 하나의 교육훈련값으로 통합하지 않고 각각의 시간으로 분석하여 한 개인의 교육훈련시간을 도출하기에는 한계가 있었다. 또한, Macduffiem & Kochan(1995)의 연구에서는 입직 이후 이수한 훈련시간을 입직 후 6개월 이내 훈련과 최근 1년 사이 훈련으로, OJT, Off-JT 형태/ 관련 직무와 관련 직무 외 훈련을 모두 구분하여 측정하였으나, 학력, 전공 등을 예, 아니오로 응답하게 하여 집단을 세분화하는데 한계가 있었다.

전공을 교육훈련에 고려한 연구들을 살펴보면, 학력이나 교육기간보다는 교육훈련 내용과 직무와의 연계성을 강조한 것이라고 볼 수 있다. Mieg(2007)과 윤형한(2008)은 고등교육에서의 전공여부를 예, 아니오로 측정하였으나, 이 방법은 고등교육에서의 학교급 수준을 고려하지 못하는 한계가 있었다.

〈표 II-18〉 선행연구에서의 교육훈련 측정방법

구분	Chi, Claser & Farr(1988) 김난영 & 조원혁(2012)	박지윤(2010)	Macduffiem & Kochan(1995)	Mieg(2007)	윤형한(2008)	노임단가등급	건설기술자 역량지수	
대상	-	15세 이상 취업자	자동차 생산 공장 종사자	환경전문가	직업상담가	소프트웨어 기술자 전체 종사자	건설기술자	
용어	학위	학교교육	훈련(training)	전공여부	상담 관련 교육	학력	학력지수	학교교육
정의 및 특징	개인이 보유한 최종 학위 (degree obtained)	교육년수	입직 이후 이수한 훈련시간	고등교육에서의 전공여부	· 학력 · 전공 · 교육참여(훈련)	학력	직무 관련 분야 내에 학력이 둘 이상인 경우 그 중 배점이 높은 학력의 배점을 의미함.	최근 3년 이내 관련 교육 참여시간
측정	최종학력	정규교육기간 (year)	입직 후 6개월 이내 훈련과 최근 1년 사이 훈련으로, OJT, Off-JT 형태/ 관련 직무와 관련 직무 외 훈련을 모두 구분하여 훈련시간으로 측정함	예, 아니오	학력은 학사와 석사 이상으로 구분하였으며, 전공의 경우, 상담 관련 전공 예, 아니오로 측정하였음. 직무교육의 경우, 참여여부(예, 아니오), 참여횟수(5회 미만, 5-9회, 10-14회, 15회 이상으로 구분)	최종학력	20점 만점 중 학사 이상 20점, 전문학사 3년제, 19점, 전문학사 2년제 18점, 고졸 15점, 비전공자 10점. 단, 석사 학위 1.5점, 박사 3점 가산점 부가	35시간 마다 1점, 최대 3점

〈계속〉

구분	Chi, Claser & Farr, (1988)	박지윤(2010)	Macduffiem & Kochan(1995)	Mieg(2007)	윤형한(2008)	노임단가등급	건설기술자 역량지수	
한계	측정하기 용이하고 교육훈련을 측정하는 대표적인 방법이며 학교급을 구분할 수는 있으나, 전공, 비정규교육 등에 대한 요소가 고려되지 않음.	측정하기 용이하나, 정규교육의 학교급별 수준, 전공 여부, 정규교육 외 비정규교육 등의 요소가 고려되지 않음.	교육훈련형태, 직무와의 연관성을 고려하여 살펴본 것은 의미가 있으나, 이를 하나의 교육훈련값으로 통합하지 않고 각각의 시간으로 분석하여 한 개인의 교육훈련시간을 도출하기에는 한계가 있음.	고등교육에서의 학교급 수준은 고려하지 않음	학력, 전공 등을 예, 아니오로 응답하게 하여 집단을 세분화하는데 한계가 있음.	측정하기 용이하고 교육훈련을 측정하는 대표적인 방법이며 학교급을 구분할 수는 있으나, 전공, 비정규교육 등에 대한 요소가 고려되지 않음.	학교급간에 따라 다른 가중치를 적용한 것은 다른 연구에 비해 우수하나, 가중치 비율 차이에 대한 타당화가 미흡함.	35점이라는 기준이 불명확하며, 전공 훈련 외 리더십 등 다른 영역에 대한 반영이 없음.

나. 자격

1) 개념 및 전문성과의 관계

표준국어대사전에서는 자격(資格)을 ‘일정한 신분이나 지위를 가지거나 일정한 일을 하는데 필요한 조건이나 능력(국립국어원, 2014)’으로 정의한다. CEDEFOP(2011) 용어집에 따르면 ‘Qualification’을 2가지로 구분하여 정의하였다. 첫 번째로는 OECD가 정의한 ‘공식적인 의미의 자격(formal qualification)’으로, “개인이 주어진 기준에 따라 학습 성과를 성취하고 특정 직업영역에서 직무를 수행하는데 필요한 능력을 보유하였을 때 획득할 수 있는 평가 및 검증과정의 공식적인 결과(자격증, 학위 등)”이며, 또다른 정의는 ILO가 정의한 ‘직무의 필수요건(job requirement)’로 “특정 직업에서 구체적인 작업을 수행하는데 필요한 지식, 기술 및 재능”으로 정의될 수 있다고 제시하였다. 또한, 자격기본법에서는 직무수행에 필요한 지식, 기술, 소양 등의 습득 정도가 일정한 기준과 절차에 따라 평가 또는 인정된 것으로 자격을 정의하고 있다. 이처럼 해당 분야의 합의하에 인증여부가 결정되기 때문에 자격은 외부 노동시장과 연계하는 신호기제로 많이 활용된다(구자길, 강순희, 2011).

이처럼 자격의 개념은 국가, 기관 및 연구자에 따라 조금씩 상이하게 정의되고 있다. 이외에도 Wirtschaftes-Lexikon(1998)는 개인이 특정 업무기능을 수행할 수 있는 능력 및 지식으로 학력 및 사전경험학습 결과 인정 등을 포함한 광의의 개념으로 정의하였으며, 이동임(2005)는 “~~~을 할 수 있는 능력”으로 일정한 기준과 절차에 따라 평가 또는 인정된 능력(지식, 기술, 및 소양 등)이라고 정의했다. 구자길 외(2007)은 노동시장에서의 일정한 역할 모형을 준거로 삼아 평가를 실시하고 그 평가에 대한 일정한 공증을 제공하는 과정으로 다른 연구자들에 비해 협의의 개념으로 자격을 정의하였다.

이를 통해 종합하면, 자격은 크게 3가지 의미로 활용된다. ① 어떤 신분이나 지위로 ‘국민의 자격, 교수의 자격으로 참석하다’와 같은 특정 신분이나 ‘어업권 행사의 자격’에서와 같이 특정 지위로 활용되고 있다. 그리고 ② 특정 신분이나 지위 달성을 위해 필요한 능력이나 자질로 정의하여 국내에서는 응시자격, 임용자격, 자격기준, 자격요건 등 특정한 요건이나 기준을 의미하는데 활용되거나 박사학위에 준하는 자격 등 포괄적인 능력이나 자질로 활용되기도 한다. 또한 ③ 능력이나 자질을 갖추고 있음을 공식적으로 인정하는 평가 및 인증 결과로 정의하여 국내에서는 증서, 증명서, 자격증, 면허증, 인증서, 수료증, 학위, 졸업장, 상장 등과 같은 의미로 사용될 수 있다.

〈표 II-19〉 자격의 개념별 활용과 예시

구분	활용	영문 용어	예시
1) 어떤 신분이나 지위	특정 신분	position, title	· 교수의 자격으로 참석하다 · 국민의 자격
	특정한 지위, 권한	right	· 어업권 행사의 자격
2) 특정 신분이나 지위 달성을 위해 필요한 능력이나 자질	특정한 요건, 기준	requirement	· 응시자격, 임용자격, 자격기준, 자격요건
	포괄적 능력, 자질	capacity, competency	· 박사학위에 준하는 자격
3) 이러한 능력이나 자질을 갖추고 있음을 공식적으로 인정하는 평가 및 인증 결과	자격증	certificate	· 국가기술자격, 민간자격 등
	면허증	license	· 의사면허 운전면허
	인증서	accreditation	· 품질인증서, 우수기업인증서
	수료증	certificate	· 교육수료증, 이수증
	학위	degree	· 학사학위, 석사학위, 박사학위
	졸업자(증)	diploma	· 고등학교 졸업장(증)
	상장	award	· 우등상 등 상장

주) 이승일, 강순희, 장현진(2012). 국가기술자격 운영 실태 및 현황 분석. 고용노동부.

자격은 특정 직무 또는 분야의 업무를 수행할 수 있는 일정 수준의 숙련이나 자질을 갖추고 있음을 공식적으로 인정하는 것이므로 전문성 및 능력 수준을 측정하는 연구 및 제도에서 중요한 변인으로 사용되고 있다. 특히 우리나라의 경우, 국가기술자격에 대한 신뢰가 높아 다수의 제도에서 이를 활용하고 있다. 한 예로, 소프트웨어 산업 기술자들의 등급체계인 ‘소프트웨어 기술자의 노임대가’에서도 등급을 구분하는 지표로 국가기술자격증 활용하고 있으며, 법령으로 지정된 ‘건설기술자의 등급 및 경력인정 등의 기준’에서도 현장에서 활용 중인 능력을 책정하는 기준으로 관련 자격을 활용하고 있다(「소프트웨어산업 진흥법 시행령」 제16조, 「건설기술 진흥법 시행규칙」 제18조제10항). 그 외 신희이(2011), 백혜림(2013)의 연구에서는 관련 자격증의 보유 여부와 전문성의 관계에서 정적 상관을 보였으며, 감사인력의 전문성을 본 김난영, 조원혁(2012)의 연구에서도 관련 자격증인 변호사, 회계사 등의 자격증 보유 여부를 고찰하기도 하였다.

2) 측정방법

자격에 대한 측정방법을 살펴보면 공식적인 의미의 자격으로 활용될 때에는 국가기술자격, 민간자격 등 자격증을 기준으로 측정하며, 직무의 필수요건 의미의 자격으로 활용될 경우에는 해당 분야의 실무 경험 등을 기준으로 측정한다.

공식적인 의미인 자격증으로 측정한 연구 중 가장 용이하게 측정한 방법은 관련 자격증의 보유 여부를 묻는 것이다. 대표적인 연구로 윤형한(2008)의 연구는 직업상담사의 자격증을 측정하기 위해 ‘한국산업인력공단 직업상담사 자격(1급, 2급)을 취득하셨습니까?’ 단일문항에 예, 아니오로 응답하게 하였다. 이 방법은 자격증의 수준과 개수를 고려하지 못한다는 한계를 갖는다. 이러한 한계를 극복하기 위해 임현우와 오석태(2014)의 연구는 조리전공 대학생의 자격증을 측정하기 위해 0개부터 개인이 보유한 조리 관련 자격증 개수를 기재하는 방법을 활용하였다. 이 방법은 자격증의 개수를 반영할 수는 있으나 자격의 수준을 고려하지 못한다는 한계를 갖는다. 자격증의 수준과 개수를 반영하기 위해 김현진(2005), 건설기술자 역량지수는 이 두 가지 요소를 고려하여 하나의 지수로 표현하였다. 김현진(2005) 연구에서는 금형제작에 관련된 자격증을 선정한 후, 수준별 가중치를 부여하여 지수화하기 위해 각 자격증이 요구하는 실무경험과 기능의 수준을 고려하여 기술사 10점, 기능장 8점, 기사 4점, 산업기사 3점, 기능사 1점으로 부여하고, 보유자격의 개수와 자격별 측정척도를 곱하여 지수화하였다. 하지만 기사자격을 2개 보유하여 도출된 8점과 기능장 자격 1개를 보유하여 도출된 8점이 현장에서 같은 자격수준을 의미하지 않는 한계가 나타났다. 또한, 건설기술자 역량지수에서의 자격지수는 취득한 국가자격의 직무 및 전문분야별로 동일한 분야 내에 취득한 국가자격이 둘 이상인 경우 그 중 배점이 높은 자격종목의 배점에 따르도록 하였으며, 40점 만점 중 기술사 40점, 기사 및 기능장 30점, 산업기사 20점, 기능사 15점 기타 10점으로 측정하였다. 이러한 시도는 자격 수준에 따른 가중치를 반영한 것은 우수하나, 수준간 가중치 비율에 따른 타당성이 부족하다는 한계를 보였다.

〈표 II-20〉 선행연구에서의 자격 측정방법

구분	Reuber & Fischer(1994)	김현진(2005)	임현우, 오석태(2014)	윤형한(2008)	노임단가 등급	건설기술자 역량지수
대상	바이오테크놀로지 및 텔레콤 기업 종사자	금형산업 근로자	조리전공 대학생	직업상담사	소프트웨어 기술자 전체 종사자	건설기술자
용어	경험	자격증 보유 지수	자격증 수	자격여부 상담 관련 경험	자격증	자격지수
정의 및 특징	<ul style="list-style-type: none"> · 입직 전 경험 · 관련 산업 경험 · 관리 관련 경험 · 작은 회사 경험 · 큰 회사 경험 	금형제작에 관련된 자격증 을 선택한 후, 수준별 기중 치를 부여하여 지수화	조리 관련 자격증 보유 갯 수	<ul style="list-style-type: none"> · 직업상담사에게 요구되는 자격 보유 여부 · 상담 관련 경험보유 여부 	개인이 보유하고 있는 국가 기술자격의 수준	자격지수는 취득한 국가자 격의 직무 및 전문분야별로 동일한 분야 내에 취득한 국가자격이 둘 이상인 경우 그 중 배점이 높은 자격종 목의 배점에 따름.
측정	경험을 구성하는 입직 전 경험을 제외한 4가지 하위 요소에 대해 예, 아니오로 응답하게 하였으며, 입직 전 경험은 경험 횟수로 응 답함	각 자격증이 요구하는 실무 경험과 기능의 수준을 고려 하여 기술사 10점, 기능장 8점, 기사 4점, 산업기사 3 점, 기능사 1점으로 부여하 고, 보유자격의 개수와 자 격별 측정척도를 곱하여 지 수화함	0개부터 개인이 보유한 자 격증 개수를 기재	<ul style="list-style-type: none"> · ‘한국산업인력공단 직업 상담사 자격(1급, 2급)을 취득하셨습니까?’ 단일 문항에 예, 아니오로 응 답 · 관련 경험인 인력은행 및 상담프로그램 참여 여부 는 있다, 없다고 측정함. 	국가기술자격의 등급을 기 준으로 기술사, 기사, 산업 기사, 기능사의 수준체계를 두고 등급과 매칭함	40점 만점 중 기술사 40점, 기사 및 기능장 30점, 산업 기사 20점, 기능사 15점 기 타 10점으로 측정함.

〈계속〉

구분	Reuber & Fischer(1994)	김현진(2005)	임현우, 오석태(2014)	윤형한(2008)	노임단가 등급	건설기술자 역량지수
한계	직무경력을 관련 산업에 대한 경험, 관련 직무에 대한 경험을 구분하여 측정함. 단, 예/아니오의 응답은 직무경력의 수준을 측정하기에는 한계가 있음.	기사자격을 2개 보유하여 도출된 8점과 기능장 자격 1개를 보유하여 도출된 8점이 같은 자격수준을 의미하지 않음.	자격의 수준을 고려하지 않음.	1, 2급 자격 수준의 차이를 고려하지 않았으며, 관련 경험의 경우에도 질적인 측면이 고려되지 않음	국가기술자격 외 국가공인 민간자격, 해외자격도 있으나 국가기술자격에만 제한적으로 적용함	자격 수준에 따른 가중치를 반영한 것은 우수하나, 수준간 가중치 비율에 따른 타당성이 부족함.

다. 직무경력

1) 개념 및 전문성과의 관계

직무경력(work experience)은 한 개인이 속한 특정 직무성과에 영향을 미칠 수 있는 경험의 총체를 의미한다(Quinones, Ford & Teachout, 1995). 단순히 직무기간(tenure)만을 의미하는 횟수, 기간의 양적측정에서 나아가 경력의 복잡성, 도전성, 다양성의 질적인 측면과 강도, 타이밍의 복합적 측면으로 다차원적 관점에서 접근되고 있다.

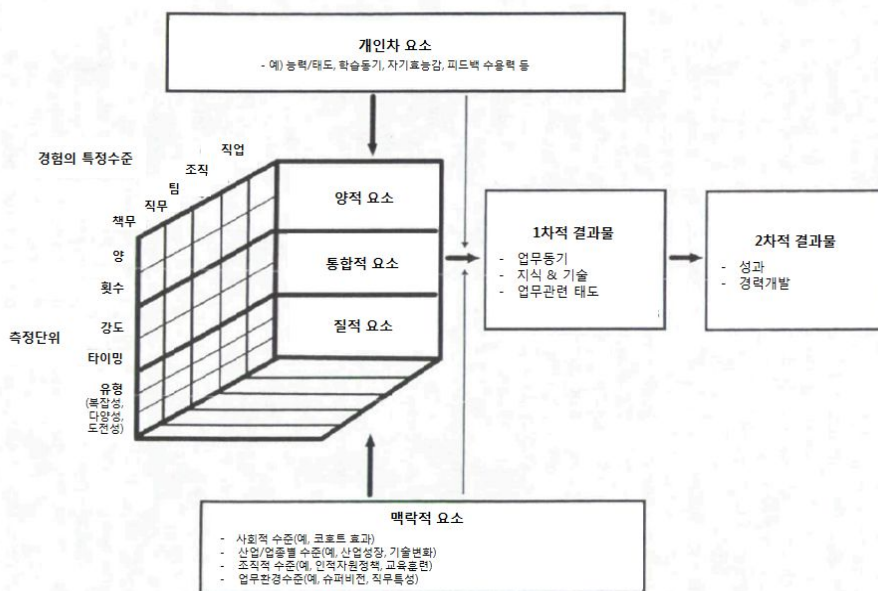
직무경력은 인적자원개발, 훈련, 경력개발 등의 연구주제에서 중요한 핵심 변인으로 언급되고 있다. 특히 Dewey(1983)가 경험학습이론을 언급하면서 경험을 통해 학습이 이루어진다는 기저를 제시하고, 성인교육의 대표적인 학자 Knowles(1950)는 성인학습자의 학습형태는 형식학습보다 무형식학습(informal learning)을 통해 이루어진다는 사실이 언급되면서 경험에 대한 연구는 점차 확대되었다(문세연, 2010; Enos et al, 2003). 또한 1980년대 이미 직무경력을 통한 학습이 실제 업무성과에 유의미한 상관이 있다는 메타분석 결과들이 제시되면서 경험에 대한 중요성은 더욱 부각되었다(Hunter & Hunter, 1984; McDaniel, Schmidt & Hunter, 1988). Hunter & Hunter(1984)의 직무경력과 업무성과의 메타분석 결과 이 두 변인의 상관은 .18로 나타났으며, McDaniel, Schmidt & Hunter(1988)의 연구에서는 .32의 높은 상관결과가 나타났다.

하지만 초기 직무경력의 연구들은 단순히 직무경력을 책무(task)수행 기간을 기준으로 측정하거나(Lance, Hedge & Alley, 1989; Vance, Coovert, MacCallum & Hedge, 1989), 직무(job)수행기간만으로(McCall et al., 1988) 측정하였다. 이후에도 학자들마다 직무경력을 상이하게 측정하면서 직무경력의 개념 및 측정에 대한 일관성 부족 문제가 제기되기 시작했다. 따라서 이후 Schmitt & Cohen(1989)는 실증연구를 통해 책무(task) 중심의 경험기간과 직무(job) 중심의 경험기간에 따라 나타난 성과가 매우 상이하게 다른 결과가 나타난다는 것을 증명했다(Ford, 1992).

이러한 결과는 직무경력이 복잡한 개념이며 한 차원으로 접근할 것이 아닌 다차원적으로 접근해야한다는 관점을 제시하게 되었으며, 이를 토대로 Quinones, Ford & Teachout(1995)는 직무경력(성과)의 관계에 대한 개념을 메타분석을 통해 리뷰함으로써 직무경력 측정에 대한 개념적 틀을 제시하였다.

Quinones et al.(1995)는 직무경력을 경험의 특정수준(the level of measurement specificity)과 측정단위(measurement mode)의 이차원으로 보았으며(Klein, Danserau &

Hall, 1994; Ostroff & Ford, 1989; Rousseau, 1985), 경험의 특정수준은 조직(organization), 직무(job), 책무(task)의 3수준과 측정단위는 횟수(amount), 기간(time), 유형(type)의 3수준으로 구분하여 총 9개 카테고리 스키마(3x3 categorization scheme)를 도출하였다.



[그림 II-5] 직무경력 측정의 개념적 틀

자료: Quinones, M.A., Ford, J.K., & Teachout, M.S.(1995). The relationship between work experience and job performance: A conceptual and meta-analytic review. *Personnel Psychology*, 48, 887-910.

경험의 특정수준에서 의미하는 조직은 회사단위를 의미한다. 즉, 동종업계 기준 재직 한 회사의 수, 근속기간 등을 측정하고, 직무는 구매, 영업, 프로그래머 등의 수준을 의미하며, 책무는 원가산출방법, 세금계산서 발행 등 작은 단위의 실제 업무를 의미한다. 메타분석 결과, 조직수준을 접근한 연구물은 18.2%, 직무수준은 68.2%, 책무수준은 13.6%로 나타났다. 측정단위는 관련 조직, 직무, 책무를 몇 번(the number of) 경험했는지에 대한 측정이며, 기간은 몇 개월, 몇 년 등을 종사했는지, 유형은 얼마나 복잡하고, 어렵고, 중요한 일을 했는지에 대한 질적인 부분이다. 메타분석 결과 횟수(11.4%), 기간(79.5%), 유형(9.1%)로 나타났다.

Quinnones et al.(1995)의 연구는 직무경력에 대한 다차원적인 기준을 토대로 새로운 직무경력의 카테고리를 만들었다는 점에서 매우 의미가 있다.

직무경력을 연구한 선행연구들의 결과에 따르면 다양한 직무경력은 통해 인지하지 못한 사이에 습득하게 된 암묵지(tacit knowledge)형성에 정적인 영향을 미치고, 정규교육에서 습득한 이론적 지식보다 현장에서 업무 성과에 더 큰 영향을 미친다(Sternberg & Caruso, 1985; Wanger & Sternberg, 1985; Reuber & Fischer, 1994). Reuber & Fischer(1994)는 매니저들을 대상으로 전문성의 한 구성요소로 언급되고 있는 암묵지와 경험의 관계를 연구하였다. 서로 다른 수준의 경험들을 한 매니저들에게 동일한 암묵지 측정 실험을 시행한 결과, 매니저들이 경험한 수준에 따라 성과에 유의미한 차이가 있는 것으로 나타났다.

이후 Tesluk & Jacobs(1998)는 Quinnones et al.(1995)의 다차원적 직무경력 측정의 개념적 틀을 토대로 직무경력의 이론적 모델을 제시하였다. 직무경력 모델은 Quinnones et al.(1995)의 9개 개념적 틀을 기초로 하여 경험의 특정수준 영역에서는 팀단위(work group), 직업(occupation)을 추가하였으며, 측정단위 영역에서는 강도(density), 타이밍(timing)을 추가하여 직무경력을 양적요소, 질적요소, 통합적요소로 구분하였다. Tesluk & Jacobs(1998)는 새로 추가한 통합적 요소를 부각시켰는데, 강도란 발달을 촉진시키는 정도를 의미하는 단어로 같은 1년을 일하였다하더라도, 누군가가 한 경험은 본인의 성장에 큰 영향을 미친 반면, 누군가의 경험은 상대적으로 적은 영향을 미치는 경험을 할 수 있기에 강도라는 새로운 요소를 추가하였다. 또한, 또 하나 추가한 개념인 타이밍은 장기적인 경력개발 관점에서 보았을 때, 해당 경험이 경력개발의 전 과정에서 얼마나 적절한 필요시점에서 발생하였는지의 정도는 측정하는 개념이다. 예를 들어, 신입사원시절에 경험해야 할 일들이 있는 반면, 관리자급에서 경험해야 할 일들이 있기 때문에 경험이 양적, 질적으로 우수하더라도 개인에게 적절한 시기에 기회가 주어졌는지는 이후 성과에 영향을 미친다고 보았다.

위에서 언급하였듯이 Quinnones, Ford & Teachout(1995)는 경험의 특정수준(level of Specificity)으로 책무(task), 직무(job), 회사(organization)의 3차원을 제시하였으며, Tesluk & Jacobs(1998)는 Quinnones, Ford & Teachout(1995)의 모델을 보완하여 특정수준을 책무(task), 직무(job), 팀(work group), 회사(organization), 직업(occupation)으로 구분하였다. 어떤 수준의 경험이어도 전문성 발달에 영향을 미치는지는 전문성 발달 단계 상의 개인의 현 위치에 따라 상이하다. 발달 단계상 입문자(pre-student), 초급자(beginner), 고급입문자(advanced beginner)(Dreyfus & Dreyfus, 1986)과 같이 5년 이하의 경력자의 경우 책무수준의 경험(task experience)이 전문성에 정적인 영향을 미치는 것으로 나타났으나, 관리자의 경우 유의하지 않게 나타났다. 관리자 직급에서 유의미하게 나타난 경험은 오히려

려 유사한 직무경력(job experience)와 해당 산업의 종사기간을 확인할 수 있는 회사경험(organizational experience)가 유의하게 나타났다(McEnre, 1988; Morrison & Brantner, 1992).

다양한 경험이 전문성 발달에 영향을 미친다는 여러 선행연구 들의 공통적인 결과들을 살펴보면, 전문성을 발달시키기 위해서는 짧게 다양한 경험을 하는 것과 길게 소수의 몇몇 경험을 하는 것 중 어느 것이 더 효과적인 것인지에 대한 물음이 생긴다. 이에 대해 McCall, Lombardo & Morrison(1988)은 실험결과를 통해 짧게 다양한 경험을 한 집단의 전문성이 더 높다는 결과를 제시했다. 즉, 여러 역할과 다양한 사람들을 만나고, 높은 스트레스 상황, 풀기 어려운 과제 등을 접하며 개인의 능력을 재구조화하여 문제를 해결해나가는 경험들을 통해 개인의 전문성이 향상된다는 것이다. 따라서 그 경험이 긍정적이건 부정적인 여러 상황에 직면했던 경험 속에서 해당 상황을 이해하기 위해 재해석하고, 살아남는 과정들은 경험을 통해 얻은 가치있는 학습이라는 것이다(Bruno & McQuarrie, 1992; Reuber & Fisher, 1993; Reuber, 1997).

2) 측정방법

직무경력에 대한 측정방법을 살펴보면, 개인이 입직 후 겪은 경험을 측정하기 위해 일반적으로 총 재직기간을 활용한 연구가 가장 많았다(Dyke, Fishcer & Reuber, 1992; Mieg, 2007; 윤형한, 2008).

Dyke, Fishcer & Reuber(1992)의 연구에서는 재직년수만으로 직무경력을 측정하기에 한계가 있기 때문에 직접적인 이전 관리 경험의 년수, 경험한 직무 수 및 간접적인 경험의 년수 및 직무 수에 대해 기업 규모도 구분하여 다차원적으로 측정하였다. 또한, 건설기술자 역량지수에서는 경력지수는 건설기술자가 수행했던 건설공사업무의 책임정도를 반영하여 직무경력을 측정하기도 하였다.

〈표 II-21〉 선행연구에서의 직무경력 측정방법

구분	Dyke, Fishcer & Reuber(1992)	Mieg(2007)	윤형한(2008)	노임단가 등급	건설기술자 역량지수
대상	5개 산업군의 관리자	환경전문가	직업상담가	소프트웨어 기술자 전체 종사자	건설기술자
명칭	경험유형	현장종사경험	상담 관련 경험	종사년수	경력지수
정의 및 특징	관리와 관련된 이전 경험	년수	재직기간		경력지수는 건설기술자가 수행했던 건설공사업무의 책임정도를 산정한 값
측정	직접적인 이전 관리 경험의 년수, 경험한 직무 수 및 간접적인 경험의 년수 및 직무 수에 대해 기업 규모도 구분함.	-	재직기간은 입직 후 종사년수로 측정		참여기술자의 경우 1.0, 분야별 책임기술자는 1.1, 사업 책임기술자는 1.3의 가중치를 주고 경력참여일로 산정함.
한계	직무경력을 실제적인 경력년수와 수행한 직무의 개수로 이분화하여 측정한 것은 우수하며, 기업 규모에 따라 경험의 질이 달라지는 것도 반영함. 단 이들에 대한 종합적인 하나의 값이 아닌 각각을 세분화하여 활용함.	일반적으로 직무경력을 측정하는데 가장 많이 활용되고 있는 측정방법이나, 다차원적인 직무경력을 측정하는데에는 한계가 있음.	관련 경험의 참여 여부를 예, 아니오로 측정하는데에는 직무경력의 수준을 측정하는데 한계가 있음.		경험의 책임정도에 따라 경험의 질을 구분하여 가중치를 부여한 것은 우수하나, 가중치 산정비율에 대한 타당성 확보가 미흡함.

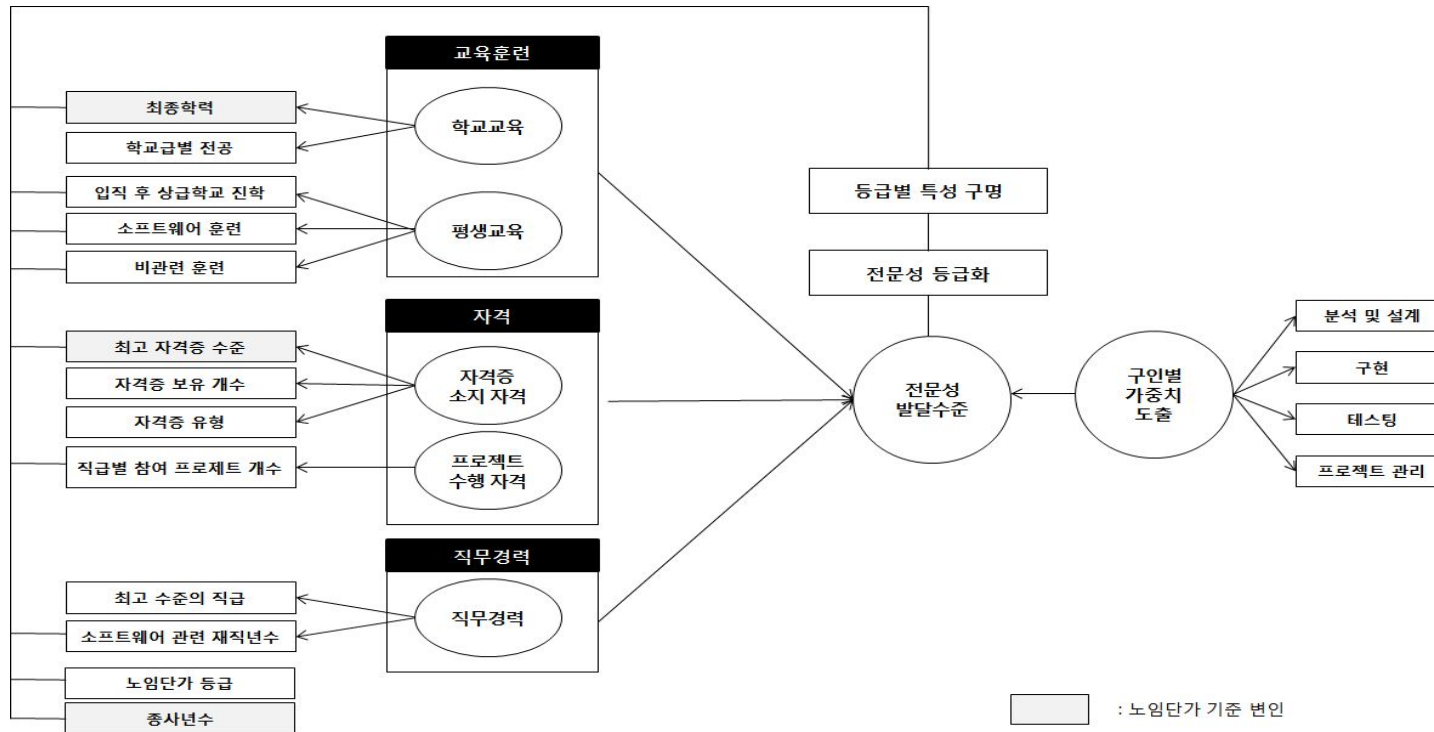
III. 연구 방법

1. 연구모형

소프트웨어 개발자의 전문성 발달수준과 교육훈련, 자격 및 직무경력의 관계를 구명하기 위한 연구모형은 [그림 III-1]과 같다. 우선 전문성 구성요인인 분석 및 설계, 구현, 테스트 및 프로젝트 관리의 가중치를 측정한 후 소프트웨어 개발자의 전문성 발달 수준을 구명한다. 소프트웨어 개발자의 전문성 발달 수준에 따라 등급을 구분한 후 최종학력, 입직 후 상급학교 진학, 소프트웨어 및 비관련 훈련 등의 관찰치를 통해 등급별 특성을 구명한다. 소프트웨어 개발자의 전문성 발달 수준과 교육훈련, 자격 및 직무경력 변인인 학교교육, 평생교육, 자격증 소지 자격, 프로젝트 수행 자격 및 직무경력의 관계를 구명한다. 학교교육은 객관적으로 관찰가능한 최종 학력과 학교급별 전공을 고려하였으며, 평생교육은 입직 후 상급학교 진학, 소프트웨어 및 비관련 훈련을 고려하였고, 자격증 소지 자격은 최고 수준 자격과 자격증 보유 개수, 자격증 유형을 고려하였으며, 프로젝트 수행 자격은 직급별 참여 프로젝트 개수를 고려하였으며, 직무경력은 최고 수준의 직급과 소프트웨어 관련 재직년수를 고려하여 연구자가 새롭게 계산한 값으로 활용하고자 한다. 끝으로 소프트웨어 개발자의 전문성 등급 집단을 구분하는 교육훈련, 자격 및 직무경력의 판별력을 구명하고자 한다. 단, 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계와 전문성 등급을 구분하는 교육훈련, 자격 및 직무경력의 판별력 분석에서는 추가적으로 기존 현장에서 활용했던 소프트웨어 개발자의 노임단가 기준 변인인 최종학력, 최고 수준 자격증, 종사년수와의 관계를 분석하여 결과를 비교해보고자 한다.

2. 연구대상

이 연구의 대상은 소프트웨어 개발자로 소프트웨어 관련 프로젝트 관리, 업무분석 및 설계, 소프트웨어 구현, DB 구축 업무를 수행하는 자이다. 한국고용정보원의 직업사전, O*Net 등 각 국의 직업사전에서는 소프트웨어 개발자의 일반적인 자격요건을 학사이상, 관련 분야 전공자로 제시하고 있다. 이를 통해 소프트웨어 개발자는 직무수준에 따라 단순 업무를 수행하는 기능직과 복합적인 업무를 수행하는 기술직의 2가지 경력경로 중 단순 업무를 수행하는 기능직은 개발자로 지칭하기 어려움으로 이 연구에서 제외하였다.



[그림 III-1] 소프트웨어 개발자의 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계 연구모형

소프트웨어 관련 프로젝트 관리, 업무분석 및 설계, 소프트웨어 구현, DB 구축 업무를 수행하는 자로 연구대상을 선정하였기에 모집단을 추정하는데에는 한계가 있어 2012년 기준 소프트웨어 산업 종사자 전체 225,772명으로 접근하였다(미래창조과학부, 2014). 한국전자정보통신산업진흥회(2014)에 따르면 국내 소프트웨어 기업 수는 6,950개이나 이에 속한 기업명단이 공시되거나 공개된 자료로 제공되지 않아 이를 활용하는데에는 한계가 있었다. 따라서 한국소프트웨어산업협회(KOSA)에 등록된 기업 중 2014년 ‘소프트웨어 기술자 역량인정체계 구축을 위한 연구’에서 활용한 250개 기업 리스트를 활용하여 표집 대상을 선정하였다. 이 중 유의표집을 통해 20개의 기업을 선정하여 이 중 소프트웨어 개발 업무 담당자를 대상으로 연구를 진행하였다.

표본크기를 설정함에 있어 모집단의 수가 10만명 이상일 경우 표본크기가 384명 이상이면 모집단을 대표하기에도 충분하다고 보고되고 있으며(Krejcie & Morgan, 1970), 일반적으로 사회과학에서는 300명 이상이면 신뢰할만한 표본집단으로 보고 있다(류근관, 2005), 이 연구의 주된 자료 분석 방법 중 하나인 군집분석을 활용하여 유형화에 있어서도 표본 크기에 대해 특별한 기준이 없었다(선택수, 2010). 또한, 특정 분야의 종사자를 대상으로 전문성 연구를 수행한 선행연구를 참조하여 표본수를 살펴본 결과 관련 연구들은 대부분은 150~500명의 사례수를 보이고 있었다(주대진, 2010; 강지아, 2010; 고영미, 이금란, 2004). 따라서 이 연구에서는 회수율, 불성실한 응답 등을 고려하여 약 600명을 설문대상으로 선정하였다.

3. 조사도구 및 변인 측정

가. 전문성 발달 수준 측정 도구

이 연구의 목적 달성을 위한 조사도구로는 질문지를 사용하였다. 질문지는 소프트웨어 개발자의 전문성 발달 수준, 교육훈련, 자격 및 직무경력 변인 영역으로 구성하였다. 응답양식은 5점 리커트 척도 및 선택형 문항을 활용하였다.

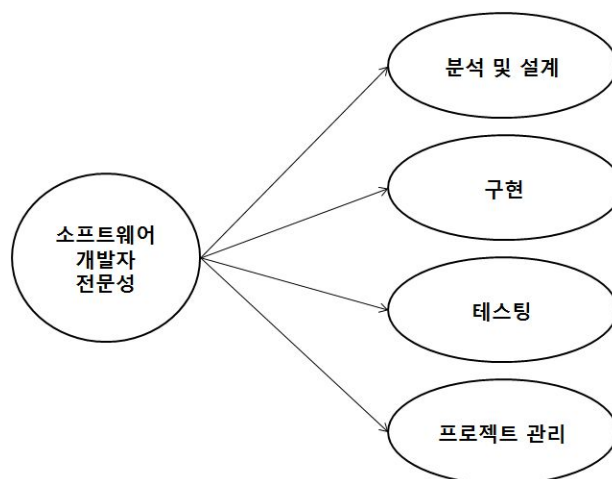
전문성 연구의 경우 자가측정으로 인한 과대측정의 오류를 범할 수 있어 전문가로 인정받은 집단과 보통 수준 집단 간의 차이를 보는 연구방법이 활용되기도 한다. 이를 위해 상급자 평가(Heijden, 2002)나 주변인 평가(Germain & Tejeda, 2012)로 진행되거나, 전문가 집단을 구분하기 위해 2인 이상의 동료 추천을 받은 자(Sonnentag, 1995; Sonnentag, 1998)로 한정하는 등의 연구방법을 진행하고 있으나, 비용과 시간의 문제

(나승일, 김기용, 2007)와 대단위 표집을 통한 양적연구가 어렵다는 한계로 인해 자기보고식 설문지를 활용한 연구도 다수 진행되고 있다(Mieg, 2007; 강지아, 2010; 허은정, 2011).

이 연구의 목적에 맞게 대단위의 표집이 이루어져야한다는 점을 감안하여 자기보고식 측정을 시행하되, 과대평가의 오류를 최소화하기 위해 Mieg(2007)이 활용한 다른 동료에게 비치는 자신의 이미지(‘다른 동료들에 비해~’)를 기준으로 측정하였다.

1) 전문성 구인화

전문성 구인을 도출하기 위해 소프트웨어 개발자의 전문성에 대한 선행연구 및 국가차원에서 제시한 소프트웨어 분야의 자격, 역량체계, 역량표준 등을 종합적으로 고찰하였다. 소프트웨어 개발자의 전문성 구인은 소프트웨어 개발 생명주기(life cycle)와 유사하게 구성된다(Sonnentag, 2006; Chilton & Hardgrave, 2004). 소프트웨어 개발자의 전문성은 고객의 요구를 분석한 후, 프로젝트 전반에 대한 큰 그림을 그리는 분석 및 설계, 프로그래밍 언어를 활용하여 구체화시키는 구현, 중간결과물 및 최종 결과물에 대한 에러를 분석하고 해결하는 테스트 및 개발능력 이외 내외부적으로 프로젝트를 이끌어 가는데 요구되는 프로젝트 관리까지 총 4가지 구인으로 구성된다([그림 III-2] 참조).



[그림 III-2] 소프트웨어 개발자의 전문성 구인

2) 전문성 발달 문항 작성

개념적 구인화 모형을 설정한 후 이를 토대로 소프트웨어 개발자의 전문성 발달 수준 측정을 위한 예비문항을 작성하였다. 우선 소프트웨어 개발자의 전문성 발달 관련 문항 풀(pool)을 작성하였으며 소프트웨어 개발자의 역량지수평가인 TOPCIT(Test Of Practical Competency IT), 한국산업인력공단에서 개발한 정보통신분야의 국가직무능력 표준(NCS)의 능력단위, 한국산업인력공단, 한국소프트웨어산업협회, 한국소프트웨어기술진흥협회에서 공동으로 개발한 소프트웨어 분야의 신자격에서 제시한 필수능력 등을 참조하여 예비문항을 작성하였다. 한편 예비문항을 작성하기 위해서는 하위영역별로 몇 개의 문항을 작성할 것인지가 먼저 논의되어야 하는데, 일반적으로 최종 검사에서 하위영역별로 3~15개 정도의 문항을 구성할 것을 제안하고 있기 때문에(Croker & Algina, 1986; Walsh & Betz, 1995), 이를 고려하여 하위 영역별로 5~15개의 예비문항을 작성하였다.

예비문항 작성은 앞서 구인화한 소프트웨어 개발자의 전문성을 Life Cycle를 근거로 하여 각 하위영역별로 5~15개의 문항을 작성하였다. 분석 및 설계 13문항, 구현 13문항, 테스트 11문항, 프로젝트 관리 6문항 총 43개 문항의 예비문항이 작성되었다. 다음으로 개발된 문항에 대하여 지시문과 응답양식을 결정하였다. 지시문은 응답자가 이해하기 쉽게 작성하였으며 이 연구에서는 지시문에 대한 측정수준으로 Dreyfus & Dreyfus(1977)이 제시한 전문성 5단계 수준을 기준으로 하되, 리커트 형태로 제시하여 지시문에 대한 정도나 빈도 아닌 해당 수준에 가장 근접한 내용을 체크하도록 하였다.

〈표 III-1〉 전문성 발달 5단계에 근거한 전문성 발달 수준 척도

수준	척도 정의
매우 잘 못함 (1수준)	지식과 기술을 배워 이론적으로 알고는 있지만 다른 동료들에 비해 실제 적용하거나 활용하는 업무를 수행하기에는 어렵다고 평가되는 수준
잘 못함 (2수준)	지식과 기술을 적용하거나 활용할 수 있어 하나의 업무가 주어지지만 다른 동료들에 비해 그 결과에 대해서는 완벽하지 않다고 평가되는 수준
보통임 (3수준)	독립적으로 하나의 업무를 맡고 있으며, 다른 동료들에 비해 업무 결과에 대해서도 신뢰받고, 상황에 맞는 의사결정과 판단이 가능하다고 평가되는 수준
잘함 (4수준)	나만의 노하우로 직관적인 업무 처리가 가능하며 다른 동료들에 비해 우수한 결과를 보인다고 평가되는 수준
매우 잘함 (5수준)	다른 동료들에 비해 판단과 결정이 무의식적으로 자동화되어 일어나며, 관련 지식과 기술의 통합적 사고가 가능하며 내가 내린 직관적 판단에 대한 비판적 성찰까지 가능하다고 평가되는 수준

3) 내용타당도 검증

연구자가 구성한 전문성 측정을 위한 문항들이 실제 소프트웨어 개발자의 전문성 수준을 측정하고 분류하는데 적합한 문항인지에 대하여 전문가 집단을 통하여 내용타당도를 검토할 것이다. 내용타당도는 소프트웨어 개발자 3인과 산업교육전공 박사 2인의 내용타당도 검토를 통해 해당 문항이 이해하기 쉽고 명확하며 기업체 상황에 적합한지, 내용이 이해하기 쉽고 명확한지 등을 검토 받을 것이다. 또한, 이 결과를 반영하여 예비문항의 초안을 수정하고 보완하여 예비조사용 최종 문항을 작성하고자 한다.

〈표 III-2〉 도구 검토 전문가

전문가 패널	선정기준
소프트웨어 분야 전문가	<ul style="list-style-type: none"> · 소프트웨어 분야 석사학위 이상 소지자 · 소프트웨어 분야 종사기간 10년 이상인 자 (단, 박사 소지자의 경우 5년 이상)
HRD 분야 학계 전문가	<ul style="list-style-type: none"> · HRD 관련 박사학위 소지자 · 관련 연구 경력자로 현재 대학 또는 연구기관 근무 · 관련 연구 주제에 대해 연구논문 발표 및 관련 분야 강의 경력 보유자

4) 예비조사

소프트웨어 개발자의 전문성 발달 수준 도구의 타당도와 신뢰도를 검증하고 본 조사를 위한 설문도구를 확정하기 위해 예비조사를 실시하였다. 초급 62명, 중급 30명, 고급 8명으로 수집된 100개의 응답을 예비조사로 분석하였으며, 내적 일치도 계수, 문항-전체 상관, 요인분석을 실시하고, 예비조사 결과에 따라 예비문항을 수정·보완하여 본 조사를 실시하였다.

신뢰도 분석결과 문항 전체의 내적일치도 계수는 .983으로 매우 높게 나타나 연구에 적합한 것으로 나타났다. 영역별로는 분석 및 설계는 .963, 구현은 .952, 테스트는 .946, 프로젝트 관리는 .943으로 매우 높게 나타났다. 다만, 문항별로 분석 및 설계 3번 문항(.611), 구현 2번 문항(.680), 테스트 10번 문항(.585), 테스트 11번 문항(.637)의 신뢰도가 낮게 나타났으나, 문항을 제거해도 신뢰도 상승 수준이 미비하여 테스트 10번 문항을 제외하곤 타당도 분석 결과를 고려하여 최종적으로 수정 및 보완을 결정하였다([부록 1] 참조).

탐색적 요인분석 결과 총 4개의 구인이 추출되었으며, 구인별 요인 적재량은 분석 및 설계 .530~.886, 구현 .551~.750, 테스트 .601~.760, 프로젝트 관리 .670~.717로 나타났다. 테스트와 프로젝트 관리 구인은 탐색적 요인분석 결과, 높은 수준의 요인적재량이 나타났으나, 분석 및 설계와 구현은 일부 문항의 경우 낮은 요인적재량이 나타났다. 탐색적 요인분석을 통해 전문성 발달 수준의 구인이 명확하게 구분되지 않아 확인적 요인분석을 추가적으로 실시하였다. 확인적 요인분석 결과, 모형적합도($\chi^2 = 8,838.91$ (df=860, $p=.000$), CFI=.973, TLI=.919, NFI=.972, IFI=.973, RMSEA=.180)가 대부분 적합하거나 양호한 것으로 나타나 문항이 항목별 구인에 유의미하게 적재되는 것으로 나타났다. 더욱이 이 연구에서 사용하는 소프트웨어 개발자의 전문성 발달 수준을 측정하는 문항은 소프트웨어 개발자에게 요구되는 능력을 전문가들이 추출한 것으로 통계적인 결과만으로 해석하기에는 무리가 있으므로 HRD 전문가 2인의 검토를 통해 낮은 신뢰도 수준이 나타난 테스트 10번 문항을 제외하고는 모두 문항에 투입하였다.

5) 본조사

최종적으로 수정된 문항을 이용하여 본조사를 실시하였다. 본조사 결과도 내적 일치도 계수, 문항-전체 상관, 요인분석을 실시하여 측정도구의 신뢰도와 타당도를 확인하였다.

소프트웨어 개발자의 전문성 발달 수준의 신뢰도 분석 결과 내적일치도 계수는 .923으로 높게 나타났으며, 분석 및 설계 .909, 구현 .897, 테스트 .939, 프로젝트 관리 .900으로 모두 .7이상의 높은 결과가 나타났다. 문항이 삭제된 경우에도 내적일치도 계수가 변하지 않거나 미미한 수준에서 변하는 것으로 나타났다. 전문성 발달 수준의 확인적 요인분석을 실시한 결과 모형적합도($\chi^2=6.424$ (df=2, $p=.040$), CFI=.988, TLI=.965, RMSEA=.150)는 대체적으로 양호하였다.

<표 III-3> 소프트웨어 개발자의 전문성 발달에 대한 확인적 요인 분석 결과

구분	비표준화 계수	표준화 계수	표준오차	C.R
분석 및 설계 ← 전문성 발달	1	.971		
구현 ← 전문성 발달	.855	.923	.046	18.676***
테스팅 ← 전문성 발달	.679	.852	.048	14.243***
프로젝트 관리 ← 전문성 발달	.410	.817	.032	12.709***

주 1) $\chi^2=6.424$ (df=2, $p=.040$), CFI=.988, TLI=.965, RMSEA=.150

주 2) 전체 문항에 대한 분석 결과임

하지만 이후 연구에서는 도구에 의한 값이 아니라 영역별 가중치를 반영하여 보정된 값으로 전문성 발달 수준을 활용하였다.

나. 교육훈련 변인 측정

교육훈련은 입직 전 정규교육과정에서 이수한 학교교육과 입직 후 사내외에서 참여한 평생교육으로 구분되며, 학교교육과 평생교육 변인은 관찰가능한 기초조사 결과를 토대로 변수를 계산한 값을 의미한다.

1) 학교교육

가) 학력 및 전공에 대한 기초조사

학교교육은 학령기 동안 정규교육에서 이수한 학습으로, 양적인 측면과 질적인 측면을 함께 고려해야 한다. 학습의 양적인 측면을 측정하는 방법으로는 최종학위(Chi, Claser & Farr, 1988; 김난영 & 조원혁, 2012)를 측정하거나, 교육년수(박지윤, 2010), 교육시간(Macduffiem & Kochan, 1995)를 활용한다. 연구에서는 흔히 측정 혹은 조사하고자 하는 변인을 조작적으로 정의하여 사용한다. 그렇기 때문에 교육정도를 어떻게 조작적으로 정의하느냐가 출발점일 수 있다. SW 개발자에게 있어서 학교교육을 정확히 파악하기 위해서는 이수한 교육과정 및 성취(성적)수준을 알아야 한다. 그러기 위해서는 SW 개발자에게 요구된 인재상 및 역량을 토대로 표준 교육과정을 설정하고, 그에 근거하여 측정해야 보다 정밀할 것이다. 하지만, 현직 종사자들을 대상으로 개개인의 기억에 의존하여 구체적으로 이수한 교육과정 및 성취도를 파악하기란 거의 불가능하다. 하지만, 개개인의 학력 및 SW 전공 유무를 조사하는 것은 객관적으로 쉽게 적용할 수 있다. 이 정보를 활용하여 SW 개발자 개개인의 교육정도를 수량화할 수 있다면 보다 정밀하게 개개인의 교육정도를 수량화할 수 있을 것이다.

이 연구에서는 김난영 & 조원혁(2012), 윤형한(2008), Chi, Claser & Farr(1988) 등에서 활용한 최종학력의 학교급에 따른 위계성을 재구성하여 개인의 고등학교 이후 정규교육 유형을 측정하였다. 이를 위해 고등학교급에서는 졸업유무(졸업, 중퇴), 유형(인문계고, 특성화고)를 측정하였으며, 대학교는 졸업유무(전문대 2년제 졸업/중퇴, 전문대 3년제 졸업/중퇴, 대졸 4년제 졸업/중퇴, 해당없음), 대학원 졸업유무(석사졸업/수료/재학, 박사 졸업/수료/재학, 해당없음)에 체크하도록 측정하였다.

또한, 학습의 질적인 측면은 이수한 학습내용으로 소프트웨어 개발과의 관련성을 측정하기 위해 직접적인 관련성 수준을 파악하기 위해 관련 직무와 관련 직무 외 학습을 구분한 Macduffiem & Kochan(1995)를 토대로 윤형한(2008), Mieg(2007)이 활용한 전공여부를 ‘소프트웨어 관련 학과’, ‘비관련 학과’를 묻는 방식으로 활용하였다. 더욱이, 대학교의 경우 소프트웨어 관련 전공자가 아니라도 복수전공을 하거나 별도로 소프트웨어 관련 과목을 수강한 경우가 있으므로 ‘소프트웨어 비관련 학과’, ‘소프트웨어 관련 교과를 수강한 적 있음’, ‘이수한 과목 총 학점’을 추가적으로 조사하였다.

나) 학교교육 변수 계산

국제적으로 유능한 SW 개발자의 기본 학력 요건은 대체로 대학교육으로 규정하고 있다. SW 개발자의 전문성은 SW 전공 유무에 따라 차이가 날 것이고, SW 전공의 경우에도 학교 급별에 따라서 차이가 있다고 볼 수 있다. 특히 학교 급별 수준에 따라 SW 개발 전공의 수준도 높다고 쉽게 가정할 수 있을 것이다. 이러한 점을 토대로 SW 개발자의 학력과 전공 유무에 관한 정보로부터 개개인의 학교교육 수준을 다음과 같이 2단계로 나뉘어서 수량화할 수 있을 것이다.

[공식 1]

학교교육 점수 = 입직 시 학력에 따른 점수 + 학교급별 전공 유무에 따른 가중치

첫째, 입직 시 학력을 기준으로 점수를 부여한다. 비록 학교 급별에 따라 수업연한의 차이가 있음에도 불구하고, 상급학교일수록 교육수준이 높다고 볼 수 있다. 이에 따라 졸업여부를 기준으로 고등학교의 경우 1, 전문대 2, 일반대 3, 석사 4, 박사 5로 최종 학력 수준에 따라 점수를 부여한다. 단, 이 연구에서는 개인의 정규교육을 면밀하게 보기 위해 졸업 유무가 아닌 학교급별로 중퇴, 재학, 수료 등을 상세히 체크하게 하였다. 따라서 고등학교를 중퇴한 경우 총 3년이므로 $n/3$ 의 비율만큼만 차감을 하고, 전문대학의 경우 2년제와 3년제 전체에 2를 부여하므로 전문대학의 최대 학기인 6학기를 기준으로 고등학교 졸업 1점에 전문대학 중퇴 $n/6$ 만큼을 추가하였다. 4년제 대학의 경우에는 전체가 8학기이므로 $n/8$, 석사의 경우는 일반적으로 논문을 포함하여 4학기에 졸업하므로 $n/4$, 박사과정의 경우 일반적으로 논문 작성 포함 평균적으로 8학기에 졸업하므로 $n/8$ 을 졸업학교급에 추가점수를 부여하였다.

둘째, 학교급별 SW 개발 전공유무 유형별(32가지) 상대적 중요도를 고려하여 계산된 가중치를 부여한다. 상대적 중요도의 원리는 기본적으로 상급학교일수록 보다 높은 수준으로 판단하되, 동시에 각 유형별 연구 대상의 상대적 빈도와 현장 종사자들의 의견을 반영한다는 두 가지 원리를 기본으로 하였다. 예를 들어, 한 수준에서만 SW 개발을 전

공하였을 경우는 모두 5가지 유형이 존재하는데, 즉, 고교에서 전공한 경우, 전문대에서 전공한 경우, 4년제 대학에서 전공한 경우, 석사과정에서 전공한 경우, 박사과정에서 전공한 경우가 있는데, 빈도에 상관없이 무조건 상급 수준에서 전공한 경우를 높게 판단한다. 다만, 2수준에서 전공한 경우는 모두 10가지가 있을 수 있는데, 이러한 경우도 상급 학교 전공을 우선시하되 빈도와 현장 관계자들의 의견을 참고하여 상대적 중요도를 판단한다. 예를들어, 4년제 대학이 소프트웨어 개발자의 요구되는 학력과 전공이라는 직업사전, O*Net의 내용과 현장 전문가들의 의견을 취합한 결과, 4년제 대학에서 전공을 하지 않고 석사에서 전공을 한 경우와 4년제 대학에서는 전공을 하였으나 석사에서 전공하지 않은 경우 전문성과의 관련성을 볼 때, 4년제 대학의 전공자에게 더 높은 가중치를 두어야 한다는 의견이 있었다.

32가지 유형의 우선순위가 결정되면 유형에 따른 가중치는 0.125로 설정한다. 이는 최종 학력에 따른 값의 최소값 1과 최대값 5 차이는 4점을 기준으로, 전체 32개 유형으로 나뉘어 유형의 상대적 중요도 순서에 따른 차이는 0.125로 계산된다. 만약 어떤 개발자가 모든 학교 급에서 소프트웨어를 전공하지 않았다면, 가중치는 0점이다.

다만, 최종학력에 따른 점수를 부여할 때 교육연한이 대체로 2년씩 차이가 나지만 실제 이수한 교육과정 및 이수 교과목 등이 다르기 엄밀한 것은 아님으로 주의가 요구된다. 또한, 2단계에서는 전공 유무 유형별 현재 종사자의 빈도를 바탕으로 한 판단은 미래의 SW 개발자 인재상과는 거리가 있음을 주의할 필요가 있다.

〈표 III-4〉 학교급별 전공유무 유형에 따른 상대적 중요도

순위	학교급별 전공 유무 유형					빈도 (명)	백분율 (%)	기중치	현장 의견
1	고등학교	전문대학	4년제대학	석사	박사	-	-	4	
2		전문대학	4년제대학	석사	박사	-	-	3.875	
3	고등학교		4년제대학	석사	박사	-	-	3.75	
4			4년제대학	석사	박사	4	.8	3.625	
5	고등학교	전문대학		석사	박사	-	-	3.5	
6		전문대학		석사	박사	-	-	3.375	
7	고등학교	전문대학	4년제대학		박사	-	-	3.25	●
8	고등학교			석사	박사	-	-	3.125	▲
9		전문대학	4년제대학		박사	-	-	3	
10	고등학교	전문대학	4년제대학	석사		-	-	2.875	●
11				석사	박사	2	.4	2.75	▲
12	고등학교		4년제대학		박사	-	-	2.625	
13		전문대학	4년제대학	석사		1	.2	2.5	
14			4년제대학		박사			2.375	
15	고등학교	전문대학			박사	-	-	2.25	
16		전문대학			박사	-	-	2.125	
17	고등학교		4년제대학	석사		7	1.4	2	●
18			4년제대학	석사		46	8.9	1.875	●
19	고등학교				박사	-	-	1.75	▲
20					박사	-	-	1.625	▲
21	고등학교	전문대학	4년제대학			2	.4	1.5	●
22		전문대학	4년제대학			6	1.2	1.375	●
23	고등학교		4년제대학			34	6.6	1.25	●
24			4년제대학			206	40.0	1.125	●
25	고등학교	전문대학		석사		-	-	1	▲
26		전문대학		석사		-	-	0.875	▲
27	고등학교			석사		2	.4	0.75	▲
28				석사		6	1.2	0.625	▲
29	고등학교	전문대학				8	1.6	0.5	
30		전문대학				21	4.1	0.375	
31	고등학교					8	1.6	0.25	
32	비전공자					162	31.5	0	

주) ● : 현장 의견을 통해 학교급별 순위에 비해 올라간 부분

▲ : 현장 의견을 통해 학교급별 순위에 비해 내려간 부분

2) 평생교육

가) 입직 후 진학 및 훈련에 대한 기초조사

평생교육은 지속적이며 자발적 동기로 인해 전문성을 높이거나 개인적인 이유로 지식을 습득하고자 하는 학습을 의미한다(Department of Education and Science, 2000). 일반적으로 학령기 시절의 정규교육 외 비정규교육을 의미하며, 입직 이후 개인의 선택에 의해 상급학교에 진학한 경우와 학교가 아닌 사내외에서 습득한 학습으로 평생교육을 측정하고자 하였다.

지금까지 선행연구에서는 교육정도를 측정함에 있어 학교교육에만 치중하고 있어 평생교육에 해당되는 입직 후 상급학교로의 진학이나 훈련에 대한 측정에 대한 선행연구는 많지 않았다. 일부 연구에서 관련 직무훈련의 참여, 참여한 훈련의 개수 정도로 측정하거나(윤형한, 2008), 일부의 연구에서 입직 이후 이수한 훈련시간을 조사하기도 하였다(Macduffiem & Kochan, 1995). 이러한 측정방법은 개인이 평생교육에 참여한 훈련의 정도를 정밀하게 측정하는데 한계를 가져왔고, 등간척도가 아니라 다양한 통계방법을 적용하는데도 어려움을 가져왔다.

따라서 소프트웨어 개발자의 평생교육을 측정하기 위해서는 학령기 이후 자발적으로 선택한 상급학교의 진학과 사내외에서 참여한 훈련 및 연수에 대한 개수와 시간, 소프트웨어 개발 직무와의 연관성을 다각도로 고려해야 보다 정밀할 것이다. 하지만, 현직 종사자들을 대상으로 개개인의 기업에 의존하게 구체적인 모든 훈련을 파악하기란 거의 불가능하다. 따라서 상급학교의 진학은 입직 시와 현재를 구분하여 그 차이를 비교하였으며, 사내외 교육 및 연수는 입직 이후 참여한 훈련으로 한정하고 소프트웨어 개발 관련 훈련 여부를 구분하여 지금까지 참여한 훈련 프로그램의 개수와 시간을 조사하는 것은 객관적으로 쉽게 적용할 수 있다. 이 정보를 활용하여 소프트웨어 개발자의 훈련정도를 수량화할 수 있다면 보다 정밀하게 개개인의 훈련정도를 도출할 수 있을 것이다.

훈련의 양적이 부분을 측정하는 방법으로는 윤형한(2008)의 관련 직무교육 참여 횟수를 측정하는 방법과 건설기술사 역량지수의 교육참여시간을 측정하는 방법을 활용하여 이수한 프로그램 수와 이수 시간을 모두 측정하였다. 또한, 토대로 훈련의 질적 부분은 교육과 마찬가지로 이수한 훈련내용을 측정하기 위해 소프트웨어 개발 관련 훈련과 관련 없는 리더십, 경제/문화 등의 훈련으로 구분하여 측정하였다.

나) 평생교육 변수 계산

소프트웨어 개발자의 전문성은 지금까지 참여한 훈련시간에 따라 차이가 날 것이고, 다양한 훈련들을 이수 여부, 특히 관련 직무와 연계될수록 전문성에 정적인 영향을 미친다는 선행연구 결과를 바탕으로 할 것이다. 이러한 점을 토대로 소프트웨어 개발자의 참여 훈련시간과 훈련 프로그램 개수, 소프트웨어 개발과의 관련성을 2단계로 나누어 수량화 하였다.

[공식 2]

$$\text{평생교육 점수} = \text{상급 학교 진학 점수} + \text{S/W 관련 훈련시간에 따른 점수} + \text{비관련 훈련시간에 따른 점수}$$

첫째, 상급 학교로의 진학 점수는 학교교육의 변수 계산 방법을 그대로 적용하되, 현재 학교교육 수준과 입직 시 학교교육 수준의 차이만큼을 적용한다.

둘째, 소프트웨어 관련/비관련에 훈련에 참여한 시간을 구하기 위해, 전체 참여 훈련시간을 도출하였다. 1단계에서 도출값을 그대로 활용하기에는 편차가 매우 심하므로 중위수를 토대로 급간을 구분하여 등간척도로 변환하였다. 이 연구에서도 등간척도로 변환하기 위해 조영재, 반상진(2013)의 방법을 활용하되, 표집 수를 반영하여 5개의 급간, 훈련하위집단(중위수 1~50%)의 경우 소프트웨어 관련 훈련시간 30시간 이하 181명, 훈련중하위(중위수 51~100%) 30시간 초과에서 60시간 이하 46명, 훈련중상위(중위수 101~150%) 60시간 초과에서 90시간 이하 17명, 훈련상위(중위수 151~200%) 90시간 초과에서 120시간 이하 65명, 훈련최상위(중위수 201% 이상) 120시간 초과 133명으로 1부터 5까지의 등간척도로 변환하였다.

〈표 III-5〉 소프트웨어 개발자의 소프트웨어 관련 훈련시간 변환 기준

구분	빈도(명)	백분율(%)	변환된 변인
소프트웨어 관련 훈련 시간	442	100.00	-
30시간 이하	181	40.95	0.1
30시간 초과 ~ 60시간 이하	46	10.41	0.2
60시간 초과 ~ 90시간 이하	17	3.85	0.3
90시간 초과 ~ 120시간 이하	65	14.71	0.4
120시간 초과	133	30.09	0.5

비관련 훈련시간의 경우 별도의 중위수를 사용하지 않고 소프트웨어 관련 훈련시간의 중위수 기준을 적용하였다. 그 이유는 비관련 훈련시간의 중위수가 10으로 적은 시간만 훈련을 받아도 변환된 변인으로 같은 값을 받게 되기 때문이었다. 예를 들어, 소프트웨어 관련 훈련시간의 경우 30시간을 받은 경우 0.1의 점수를 받지만, 비관련 훈련시간의 경우 5시간만 이수해도 0.1의 점수를 받게 된다. 이는 소프트웨어 훈련에 더 큰 가중치가 부여되어야 한다는 기본 가정에 반대되는 논리임으로 소프트웨어 훈련과 같은 중위수 기준을 적용하도록 하였다. 그 결과 훈련하위집단(중위수 1~50%)의 경우 훈련시간 30시간 이하 285명, 훈련중하위(중위수 51~100%) 30시간 초과에서 60시간 이하 46명, 훈련중상위(중위수 101~150%) 60시간 초과에서 90시간 이하 23명, 훈련상위(중위수 151~200%) 90시간 초과에서 120시간 이하 14명, 훈련최상위(중위수 201% 이상) 120시간 초과 72명으로 1부터 5까지의 등간척도로 변환하였다.

〈표 III-6〉 소프트웨어 개발자의 소프트웨어 관련 비관련 훈련시간 변환 기준

구분	빈도(명)	백분율(%)	변환된 변인
비관련 훈련 시간	440	100.00	-
30시간 이하	285	64.77	0.1
30시간 초과 ~ 60시간 이하	46	10.45	0.2
60시간 초과 ~ 90시간 이하	23	5.23	0.3
90시간 초과 ~ 120시간 이하	14	3.18	0.4
120시간 초과	72	16.36	0.5

다. 자격 변인 측정

자격은 특정 직업에서 직무를 수행하는데 필요한 능력을 의미하며 공식적으로 검증된 자격증의 보유 여부와 자격증이 보유하지 않아도 현장에서 겪은 경험을 통해 해당 능력이 있다는 것을 알 수 있다. 이 연구에서 자격은 자격증 소지 자격과 프로젝트 수행 자격으로 구분되며, 관찰가능한 기초조사 결과를 토대로 변수를 계산한 값을 의미한다.

1) 자격증 소지 자격

가) 소지 자격증 유형 및 수준에 대한 기초조사

자격증은 일정 수준의 도달 여부를 인증하는 대표적인 변인으로 국가기술자격, 국가공인민간자격, 해외자격으로 구분하여 측정하였다. 교육훈련 변인과 다르게 해당 분야의 능력 여부를 측정하는 것으로 소프트웨어 개발 관련 자격증으로 범위를 한정하였다.

공식적으로 검증된 자격으로 가장 많이 활용되는 지표는 자격증이다. 자격증은 국가기술자격, 민간자격이 있으며 소프트웨어 분야의 경우 국제적으로 통용되는 민간자격이 있다. 소프트웨어 개발자에게 있어 소지한 자격을 파악하기 위해서는 자격증명과 자격급수를 조사하는 것이 일반적이다.

선행연구에서는 자격증의 신뢰도 및 편의를 위해 국가기술자격으로 한정하여 측정하며 일반적으로 소지하고 있는 자격증 중 가장 높은 수준의 자격증을 조사한다. 하지만, 개인이 소지하고 있는 자격은 다양하며 그 개수도 여러 가지라 가장 높은 수준의 자격만 조사하는 것은 개개인의 소지 자격 수준 정도를 정밀하게 측정한 것이라고 보기 어렵다.

다만, 모든 자격을 조사하는 것은 거의 불가능하기 때문에 소프트웨어 개발에 관련되어 있다고 판단되는 자격을 설문지에 미리 제시하여 해당되는 자격증의 보유 여부를 조사하였으며, 자격증의 급수는 국가기술자격의 기술사, 기사, 산업기사, 기능사의 4수준을 기준으로 하였다.

이를 위해 국가직무능력표준(2014)에서 제시한 소프트웨어 개발 관련 자격증 종류를 토대로 기술사의 경우 정보관리, 컴퓨터시스템응용, 정보통신, 전자계산기, 전자응용, 전자계산기조직응용을 제시하였으며, 기사의 경우 정보처리, 전자계산기조직응용, 정보통신(산업), 정보보안, 전자계산기, 산업기사의 경우 정보처리, 사무자동화, 정보통신, 정보보안, 기능사의 경우 정보기기응용, 정보처리, 전자계산기 자격증 명칭을 구체적으로 제시

하였다. 국가공인민간자격으로는 SIS(정보보호전문가), 데이터아키텍처전문가, 정보시스템감리사, 정보보호전문가, PC 활용능력 평가시험, 정보시스템감리사, 리눅스마스터, 인터넷정보관리사, 디지털정보활용능력(DIAT), 컴퓨터운용사, PC정비사, PC Master, 정보기술자격시험(ITQ), 정보기술프로젝트관리전문가(IT-PMP)를 제시하였으며, 해외자격으로는 CISA(국제공인정보시스템 감사사), CISSP(국제공인정보시스템 보안전문가), OCP, SCJP, MCDBA, OCP-DBA, CCIE, CSA를 제시하였다. 이와 같이 구체적인 자격증명을 제시한 후 이 중 보유하고 있는 개수를 기재하도록 측정하였다.

나) 자격증 소지 자격 변수 계산

소프트웨어 개발자의 소지한 자격 수준은 자격증의 급수가 높을수록, 같은 등급에서도 보유한 자격증이 많을수록, 다양한 자격증을 보유할수록 자격 수준이 높다고 쉽게 가정할 수 있다. 예를 들어, 최고 수준으로 기사자격 1개를 보유하고 있는 사람과, 최고 수준으로 기사자격을 보유하고 있어 같은 기사자격 소지자로 보이는 사람의 세부적인 자격보유 현황을 조사하니, 기사수준의 자격증을 3개 보유하고 있고, 그 밑의 산업기사 2개, 기능사 1개의 자격을 보유하고 있다면 앞서 기사자격 1개를 보유하고 있는 사람과 동일한 자격 수준이라고 여기기에는 어려움이 있다.

따라서 최고수준의 자격 등급, 최고 등급에서의 보유 개수, 자격보유유형을 모두 반영하여 개인의 소지 자격수준을 다음과 같이 3단계로 나누어 수량화 할 수 있을 것이다.

[공식 3]

자격증 소지 자격 점수 = 최고 수준 자격 등급에 따른 점수 + 자격 보유 개수에 따른 가중치 + 자격 보유 유형에 따른 가중치

첫째, 최고 수준의 자격 등급은 국가기술자격의 4단계 등급체계를 기준으로 하였다. 기술사 수준은 4점, 기사 3점, 산업기사 2점 그리고 기능사 1점으로 자격 등급에 따라 점수를 부여한다.

둘째, 개인이 가진 최고 수준의 자격급에서 보유한 자격증의 개수를 측정한다. 다만 기사자격은 국가기술자격과 국가공인민간자격 및 해외자격의 두 부분으로 구분하여 각각

0.25의 가중치를 부여하여 측정하였다. 최고 등급에서의 보유 개수를 통한 가중치는 최대 0.5로 하여 아무리 많은 자격증을 보유하고 있더라도 상위 자격등급을 넘지 못하도록 설정하였다. 이는 산업기사를 10개 가지고 있더라도 기사 자격 1개를 보유하고 있는 사람의 자격수준보다는 낮다라는 현장의 의견을 반영한 결과이다.

국가기술자격은 최고 수준의 자격이 기사라면 기사급에 해당되는 자격증의 수가 몇 개인지 확인한 후, 소프트웨어 분야에서 관련 자격으로 제시한 전체 기사급 자격증 수를 기준으로 비율을 구한다. 예를 들어, 산업기사에 2개의 자격증을 가지고 있다면 소프트웨어 개발과 관련된 산업기사의 총 자격증 수가 4개 이므로 2/4, 0.125의 가중치가 부여된다.

〈표 III-7〉 소프트웨어 개발 관련 국가기술자격 등급별 자격증의 수

등급	세부 자격명	해당등급의 총 자격증 수
기술사	정보관리, 컴퓨터시스템응용, 정보통신, 전자계산기, 전자응용, 전자계산기조직응용	6
기사	정보처리, 전자계산기조직응용, 정보통신(산업), 정보보안, 전자계산기	5
산업기사	정보처리, 사무자동화, 정보통신, 정보보안	4
기능사	정보기기운용, 정보처리, 전자계산기	3

다만 국가공인민간자격과 해외자격은 소프트웨어 개발 분야에 매칭된다고 조사된 합의된 자료가 존재하지 않아 조사결과 보유 빈도를 파악하여 소지 개수별 가중치를 부여하였다. 국가국가공인민간자격과 민간자격의 경우 5개 이상을 가지고 있는 경우는 526명 중 극히 일부였기 때문에 1~5개로 구분하여 0.05만큼의 가중치를 부여하였다.

〈표 III-8〉 국가공인민간자격과 해외자격에 대한 가중치 변환 기준

구분		빈도(명)	백분율(%)	가중치
국가공인 민간자격	미보유	452	85.9	0
	1개	46	8.7	0.05
	2개	19	3.6	0.10
	3개	7	1.3	0.15
	4개	-	-	0.20
	5개 이상	2	0.4	0.25

〈계속〉

구분		빈도(명)	백분율(%)	가중치
해외자격	미보유	410	77.9	0
	1개	66	12.5	0.05
	2개	38	7.2	0.10
	3개	5	1.0	0.15
	4개	3	0.6	0.20
	5개 이상	6	1.1	0.25

셋째, 자격 보유 유형은 자격등급별 보유 유형(16가지)의 상대적 중요도를 고려하여 계산된 가중치를 부여하였다. 상대적 중요도의 원리는 기본적으로 높은 등급일수록 보다 높은 수준으로 판단하되, 동시에 각 유형별 상대적 빈도와 현장 종사자들의 의견을 반영한다는 두 가지 원리를 기본으로 하였다. 16가지 유형의 우선순위에 따라 유형에 따른 가중치는 0.05로 설정하였다. 자격 등급간 차이가 1이며, 동일 등급의 자격 보유 개수 비율에 따라 최대 0.5의 값을 가지므로 자격 유형을 통한 최대값도 0.5를 두어 가중치를 얻게 설정하였다.

〈표 III-9〉 자격증 중복 유형 따른 상대적 중요도

순위	자격증 중복 유형				빈도(명)	백분율(%)	가중치
1	기능사	산업기사	기사	기술사	4	0.76	0.5
2		산업기사	기사	기술사	4	0.76	0.4375
3	기능사		기사	기술사	-	-	0.375
4			기사	기술사	10	1.90	0.3125
5	기능사	산업기사		기술사	-	-	0.25
6		산업기사		기술사	-	-	0.1875
7	기능사			기술사	-	-	0.125
8				기술사	-	-	0.0625
9	기능사	산업기사	기사		14	2.66%	0.5
10		산업기사	기사		33	6.27%	0.375
11	기능사		기사		50	9.51%	0.25
12			기사		243	46.20%	0.125
13	기능사	산업기사			4	0.76%	0.5
14		산업기사			16	3.04%	0.333
15	기능사				11	2.09%	0.166
16	없음				137	26.05%	0
계						100.00%	

2) 프로젝트 수행 자격

가) 직급별 참여한 프로젝트 수행에 대한 기초조사

프로젝트 참여는 자격증과 같이 현장에서 경험한 프로젝트를 토대로 일정 수준에 도달했다고 인증하는 것으로 프로젝트에서 수행한 직무가 중요한 요인이 된다. Quinones & Teachout(1995)는 직무경험(work experience)의 개념을 정의함에 있어 횡수나 기간의 양적인 영역 외 책임성이라는 질적인 영역을 추가적으로 강조하였다. 따라서 프로젝트에 참여한 수와 프로젝트에서 개인이 수행한 역할에 대해 이차원적으로 조사할 필요가 있다. 예를 들어, 연구원들의 능력을 평가함에 있어서도 참여한 프로젝트의 개수와 함께 해당 프로젝트에서의 역할(연구책임자, 공동연구원, 연구조원 등)을 기재하도록 되어 있다.

이를 위해 개개인 참여한 프로젝트를 나열한 후 프로젝트 내에서의 본인의 역할 등을 기재하게 하는 것이 정밀한 방법이나 인터뷰가 아닌 조사연구에서 개개인에게 설문을 통해 파악하기란 거의 불가능하다. 다만 대체적으로 한 프로젝트에 다양한 직급의 종사자가 투입되고, 직급별 하는 역할이 대체적으로 유사하다는 현장 의견에 따라 직급별로 참여했던 프로젝트 수를 조사하였다. 이를 토대로 기초조사에서 프로젝트 수행 자격을 측정하기 위해 프로젝트 참여 개수를 조사하였으며, 직급별 역할의 차이를 알기 위해 사원급에서의 참여, 대리급, 과장 및 차장급, 부장 및 팀장급, 임원급의 참여를 구분하여 해당 직급별로 참여한 프로젝트 수를 측정하였다.

나) 프로젝트 수행 자격 변수 계산

자격으로서의 프로젝트 수행 자격은 다른 사람에 비해 참여한 프로젝트 수가 많고, 높은 직급에서의 참여 경험의 많을수록 프로젝트 수행 자격이 높을 것이라고 가정할 수 있다. 기초조사를 통해 직급별로 참여한 프로젝트 개수가 확보되었으므로 이 정보로부터 개개인의 프로젝트 수행 자격의 수준을 다음 2단계를 통해 계산할 수 있다.

[공식 4]

$$\text{프로젝트 수행 자격 점수} = (1 + \text{사원급에서의 참여 수 가중치}) + (1 + \text{대리급에서의 참여 수 가중치}) + (1 + \text{과차장급에서의 참여 수 가중치}) + (1 + \text{팀부장급에서의 참여 수 가중치}) + (1 + \text{임원급에서의 참여 수 가중치})$$

첫째, 프로젝트 참여시 직급에 따라 수행하는 역할이 상이하므로, 총 프로젝트 개수를 기준으로 하는 것이 아니라 직급별 참여한 프로젝트를 기준으로 구분한다. 단, 직급별로 한번이라도 해당급에서 프로젝트를 참여한 경우에는 이를 반영해야하므로 직급별로 1의 점수를 부여한다.

둘째, 직급별로 참여한 전체 프로젝트 개수에 대한 상대적 가중치를 부여한다. 상대적 가중치의 원리는 직급별로 많은 프로젝트에 참여할수록 높은 수준으로 판단하였다. McCauley 외(1994)가 관리자들을 대상으로 수행한 전문성 연구에서 직무순환을 통해 다양한 경험을 많이 한 관리자일수록 전문성이 높아지고, 그에 따른 업무성고가 높았다는 연구결과를 반영하여 프로젝트에서 수행한 역할은 상이하지만 다양한 프로젝트에 참여했다는 것은 그만큼 다루는 내용이 상이한 다양한 경험을 겪었다는 것으로 보고 프로젝트 참여 개수가 높을수록 상대적으로 더 높은 점수를 부여하였다. 단, 직급별로 많은 프로젝트에 참여했을 경우 직급별 부여된 점수에서 가중치를 추가하는 방향이 아닌 직급별 최사분위를 기준으로 차감하는 방식을 적용했다. 그 이유는 현장에서 예를 들어, 대리급의 능력을 인정받으려면 평균 이 정도는 해야한다는 의미로 2점을 부여하는 것인데, 대리급에서 참여해야 하는 평균 5.89개의 프로젝트에 참여하지 않은 것은 직급은 대리이나 사원 정도의 능력 수준을 보유한 것으로 봐야하기 때문이었다. 또한, 평균이상 참여를 많이 했다 하여 2 이상의 값을 갖는 것은 대리급에서 많은 프로젝트에 참여했다 하여 과장이 하는 일을 한다고 볼 수는 없기 때문이다.

가중치를 부여하는 방식은 평균을 중심으로 한 사분위수(25%, 50%, 75%)를 기준으로 하여 25%일 경우 -.75, 25~50%일 경우 -.50, 50~75%일 경우 -.25, 75%이상일 경우 0으로 차감되는 점수가 없게 설정하였다. 직급별 사분위수에 따른 빈도는 다음과 같다.

〈표 III-10〉 소프트웨어 개발자의 직급별 프로젝트 참여 변환 기준

구분	빈도(명)	백분율(%)	가중치
사원급 프로젝트 참여 평균 6.66	425	100.00	-
2개 이하	135	31.76	-.75
2개 초과 ~ 4개 이하	124	29.18	-.50
4개 초과 ~ 6개 이하	67	15.76	-.25
6개 초과~	99	23.29	0
대리급 프로젝트 참여 평균 5.89	327	100.00	-
2개 이하	97	29.66	-.75
2개 초과 ~ 4개 이하	81	24.77	-.50
4개 초과 ~ 6개 이하	68	20.80	-.25
6개 초과~	81	24.77	0
과차장 프로젝트 참여 평균 9.06	172	100.00	-
3개 이하	57	33.14	-.75
3개 초과 ~ 5개 이하	30	17.44	-.50
5개 초과 ~ 10개 이하	51	29.65	-.25
10개 초과~	34	19.77	0
팀부장 프로젝트 참여 평균 8.76	29	100.00	-
3.5개 이하	7	24.14	-.75
3.5개 초과 ~ 6개 이하	10	34.48	-.50
6개 초과 ~ 10개 이하	6	20.69	-.25
10개 초과~	6	20.69	0
임원급 프로젝트 참여 평균 7.00	8	100.00	-
1개 이하	3	37.50	-.75
1개 초과 ~ 5개 이하	2	25.00	-.50
5개 초과 ~ 12.25개 이하	2	25.00	-.25
12.25개 초과~	1	12.50	0

라. 직무경력 변인 측정

직무경력 변인은 관찰가능한 기초조사 결과를 토대로 변수를 계산한 값을 의미한다.

가) 직급별 경력기간 및 소프트웨어 개발 경력에 대한 기초조사

직무경력에 특정 직무성과에 영향을 미칠 수 있는 경험의 총체 정도로(Quinones & Teachout, 1995) 양적 부분과 질적 부분으로 구성하였다. 지금까지 선행연구에서 직무 경력을 측정함에 있어서 흔히 총 재직년수만을 활용하였지만, 이 때 재직년수는 사원급에서의 1년과 임원급에서의 1년을 동일하게 적용한 것으로 직급을 고려하지 않았다는 한계가 있었다. 이를 위해 일부 선행연구에서는 경험년수와 함께 경험한 직무의 수나 기업의 규모로 측정하기도 하였으며(Dyke, Fishcer & Reuber, 1992), 전체 종사년수가 아닌 해당 직무에 대한 경험년수로 한정하여 측정하기도 하였다(윤형한, 2008). 또한, 더 세부적으로는 입직 전, 해당 산업, 해당 직무, 기업 규모 등으로 세분화하여 직무경력을 측정하는 연구도 있다(Reuber & Fischer, 1994). 위의 방법을 활용하면 정밀하게 측정할 수는 있으나 개인이 거친 이력 모두를 파악해야하므로 인터뷰가 아닌 설문을 통해 조사연구를 하는 한계가 있어 어려움이 있다.

양적 부분은 Mieg(2007), 윤형한(2008), Dyke, Fishcer & Reuber(1992)의 직무년수를 활용하여 측정하였다. 질적 부분은 Quinones & Teachout(1995)가 제시하였듯이 책임성을 측정하기 위해 프로젝트 참여 변인과 마찬가지로 사원급, 대리급, 과장 및 차장급, 부장 및 팀장급과 임원급으로 구분하여 근무기간을 측정하였다. 또한, 소프트웨어 개발 분야의 근무기간과 소프트웨어 개발 분야 외 근무기간을 구분하여 조사하였다.

나) 직무경력 변수 계산

직무경력은 다른 사람에 비해 종사년수가 길고, 같은 종사년수라도 높은 직급의 경력이 있을수록 직무경력이 높을 것이라고 가정할 수 있다. 기초조사를 통해 직급별로 경력년수가 확보되었으므로 이 정보로부터 개개인의 프로젝트 수행 자격의 수준을 다음 2단계로 나누어 수량화할 수 있을 것이다.

[공식 5]

직무경력 점수 = S/W 관련 재직년수에 따른 기중치 + 개인이 도달한 최고 수준의
직급에 따른 점수

첫째, 총 재직년수에 대한 상대적 가중치를 부여한다. 상대적 가중치의 원리는 직급에 관계없이 재직년수가 길수록 높은 수준으로 판단하였다. 전문성의 대원칙이 경력이 길수록 전문성이 높아진다는 것을 반영하여 직급에 관계없이 종사한 기간이 길수록 상대적으로 더 높은 점수를 부여하였다. 단, 총 재직년수는 소프트웨어 개발과 비관련 재직기간을 통합한 전체 재직년수로 보았다. 점수를 부여하는 방식은 훈련에서 사용하였던 조영재, 반상진(2013)의 중위수를 기준으로 급간을 구분하는 방법을 적용하여 최고 수준의 직급에 의해 사원에서 임원의 점수 차이가 4점이므로 1~4까지의 등간척도로 변환하기 위해 종사년수 최하위집단(중위수 1~50%), 종사년수 하위집단(중위수 51~100%), 종사년수 중상위집단(중위수 101~150%), 종사년수 상위집단(중위수 151~200% 이상), 종사년수 최상위집단(중위수 201%)으로 총 5개 구분하였다.

〈표 III-11〉 소프트웨어 개발자의 소프트웨어 종사년수 변환 기준

구분	빈도(명)	백분율(%)	가중치
소프트웨어 종사년수(중위수 6)	448	100.00	-
3년 이하	112	21.6	1
3년 초과 ~ 6년 이하	114	22.0	2
6년 초과 ~ 9년 이하	87	16.8	3
9년 초과 ~ 12년 이하	66	12.7	4
12년 초과~	69	13.3	5

주) 결측치 (N=78) 제외

둘째, 개인이 도달한 직급이 높을수록 직무경력이 높다고 볼 수 있으므로 직급별 가중치를 부여한다. 일반적으로 사원급 1, 대리급 2, 과장 및 차장급 3, 부장 및 팀장급 4, 임원급 5로 명명척도를 등간척도화 시키기도 하나 입직 후 경험을 측정하는 목적에 따라 직급보다는 재직년수가 개인의 직무경력에 중요한 요소이므로(Dokko & Rothbard, 2009) 종사년수 1~5의 가중치보다 작도록 설정하였다. 예를 들어, 박사 학위를 소지하고 과장으로 입사한 경우에도 3점을 받고, 10년을 종사해도 3점을 받는 것은 경험을 명확히 측정한 것이라고 보는데 한계가 있었다. 따라서 사원급 0.5, 대리급 1, 과장 및 차장급 1.5, 부장 및 팀장급 2, 임원급 2.5로 가중치를 부여한다. 단, 예를 들어 임원을 역임했으나 현재 부장일 경우에는 현재 직급이 아닌 개인이 도달한 최고 수준의 직급인 임원으로 반영하였다. 단, 개인이 도달한 최고 수준의 직급은 소프트웨어 개발 분야에서의 경력으로만 한정하였다.

4. 자료수집

이 연구의 목표 모집단인 소프트웨어 개발자는 소프트웨어 산업에서 소프트웨어 구현(개발), 아키텍처, 프로젝트 관리 직무를 수행하는 자이다. 모집단 전체를 체계적으로 접근하는데 한계가 있으므로 한국소프트웨어산업협회(KOSA)에 등록된 기업 중 2014년 ‘소프트웨어 기술자 역량인정체계 구축을 위한 연구’에서 활용한 250개 기업 리스트를 활용하여 표집 대상을 선정하였다. 이 중 유의표집을 통해 20개의 기업을 선정하였으며, 선정된 기관의 관계자에게 전화연락과 이메일로 자료를 수집하였다.

자료수집은 예비조사의 경우, 전문성 측정도구의 경우 행동을 측정하는 것이 아닌 능력을 측정한다는 점에서 면밀하게 검토되어야 할 부분으로 1차와 2차에 걸쳐 진행할 예정이다. 이를 위해 한국소프트웨어산업협회 등 소프트웨어 산업 관련 협회(SC, Sector Council)의 담당자를 미리 방문하여 20개의 기업 담당자의 연락처를 확보한 후, 협조를 요청드렸다. Dilman(2000)이 제시한 자료수집 방법을 활용하여 먼저, 첫 대면을 위해 협력자에게 개별적인 연락을 취하여 연구의 취지를 설명하며, 해당 기업의 응답가능 인원을 파악한 뒤 설문지를 배포하였다.

또한 회수율을 높이기 위해 방문, 우편, 온라인 조사를 병행하여 진행하고, 중간 중간 기업별 협력자에게 연락하여 설문이 진행되는지 여부를 파악하여 회수를 독려했다. 최근에는 기업 내부의 방침으로 인해 인쇄물을 통한 설문조사가 어려운 경우가 많고 소프트웨어 분야 특성상 온라인에 익숙하여 온라인 조사를 병행하였다. 이 경우에는 서베이몽키(surveymonkey)를 활용하였으며, 응답자의 편의를 고려하여 모바일 폰으로도 배포하였다.

자료수집은 2015년 11월 2일부터 11월 13일까지 온라인 및 오프라인 설문으로 2주간 진행되었으며, 응답자의 편의성을 고려하여 온라인 설문으로 주로 실시되었다. 총 681부가 배포되어 한국소프트웨어산업협회의 소프트웨어 기술자의 노임대가등급 중 단순 직무를 수행하는 기능사에 해당되는 초급기능사, 중급기능사, 고급기능사 87부, 불성실하게 응답한 6부를 제외한 526명이 최종 분석에 활용되었다.

5. 자료분석

이 연구에서는 소프트웨어 개발자의 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계를 구명하기 위해 SPSS for Windows 21.0, AMOS 21.0 프로그램을 사용할 예정이다. 모든 분석에 있어 통계적 유의수준은 사회과학연구에서 일반적으로 설정하는 0.05로 설정하여 처리하였으며, 경우에 따라 0.01수준에서도 유의도를 판단하여 제시하였다. 연구문제에 따른 연구 방법을 보면 다음과 같다.

〈표 III-12〉 통계분석방법

연구문제		분석방법
연구문제 1	소프트웨어 개발자의 전문성 발달 수준은 어떠한가?	• 확인적 요인분석 • 주성분 분석
연구문제 2	소프트웨어 개발자의 전문성 발달 수준에 따른 전문성 등급은 어떻게 구분될 수 있는가? 등급별 특성은 어떠한가?	• 군집분석 • 빈도분석
연구문제 3	소프트웨어 개발자의 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계는 어떠한가?	• 중다회귀분석
연구문제 4	소프트웨어 개발자의 전문성 등급 집단을 구분하는 교육훈련, 자격 및 직무경력의 판별력은 어떠한가?	• 판별분석

가. 주성분 분석

이 연구에서는 소프트웨어 개발자의 전문성을 기술영역, 비즈니스 영역으로 구인화 하였으며, 이들 영역에 대한 가중치를 산정하기 위해 주성분 분석을 활용하였다. 주성분 분석(principal component analysis)은 다차원의 데이터의 변량을 최대한으로 유지하며 차원축소를 행하는 방법이다. 다차원 변인의 변량을 주성분이라는 적은 수의 변인으로 축소하는 기법이라 할 수 있겠다. 이 때 주성분 벡서를 함수로 나타낼 수 있다. 이 함수식에서 벡터값을 찾는 것이 가중치로 활용된다. 가중치를 구하는 방법으로 AHP와 함께 주로 활용되고 있다.

일반적으로 주성분 분석은 탐색적 요인분석의 결과값 중 초기 고유값의 분산값을 가중치값으로 사용하고 있다. 따라서 이 연구에서도 전문성 발달 수준 4가지 구인의 가중치를 확인하기 위해 초기 고유값의 분산%을 가중치 값으로 사용하였다.

나. 군집분석

이 연구에서는 소프트웨어 개발자의 전문성 발달 수준을 등급화 하기 위해 군집분석을 활용하였다. 군집분석은 변인이 많거나 명확한 분류기준이 없는 경우에 다양한 특성을 지닌 연구대상들의 유사성을 중심으로 분류된다. 이 때 군집분석의 결과는 동일한 집단에 속한 개체들의 특성이 유사(similarity)하며, 서로 다른 집단에 속한 개체들 간의 특성은 이질적(dissimilarity)이 되도록 각 개체를 분류하는 탐색적 통계적 자료 분석 방법이다(Lance & Williams, 1966, 1967).

군집분석은 계층적 군집분석과 k-mean 군집분석이 있다. 일반적으로 표본이 100개 이하일 경우 계층적 군집분석을 활용하며, 표본이 100개 이상일 경우 k-mean 군집분석을 실시한다(Aldenderfer & Blashfield, 1984). 다만 계층적 군집분석의 경우 군집의 수를 연구자가 정하지 않고 가장 타당한 군집의 개수로 구분되는 반면, k-mean 군집분석은 연구자가 임의로 군집의 수를 정해야 하기 때문에 별도로 군집 개수에 대한 타당도 검사를 실시해야 한다.

특히 이 연구에서 활용할 계층적 군집분석(hierarchical cluster analysis)는 관찰값들 간의 거리행렬을 기준으로 가까운 관찰값들을 서로 묶어나가는 병합(agglomeration)방법과 거리가 먼 값들을 나뉘가는 분할(division)방법이 있는데, 주로 병합방법이 활용되며 절차는 다음과 같다(배화수 외, 2008). N개의 자료를 각각 하나의 집단으로 가정하여, $N \times N$ 형태의 자료에 대한 거리행렬 d를 만든다. 이후 N개의 집단 중 가장 거리가 가까운 두 개의 집단을 병합하여 n-1개의 집단으로 군집을 축소한 후, 새롭게 생성된 집단과 나머지 집단들 간의 거리를 계산하여 새로운 d를 만든다. 이후 집단 간의 거리적도에 따라 각 단계에서의 병합과정을 총 n-1번 반복하여 모든 관찰값들이 하나의 집단에 포함되도록 한다. 끝으로 가장 적절한 집단의 수를 정한다.

다. 판별분석

종속변인이 이항 또는 다항으로 된 범주형일 때 적용할 수 있는 분석방법으로 다변량 기법인 다중판별분석(multiple discriminant analysis: MDA)이다. 판별분석은 일반적으로 집단들 간의 차이를 판별하는데 매우 유용한 도구로, 계량적 자료로 측정된 독립변인을 이용한다. 판별함수를 통해 집단분류의 기준을 보여주며 이러한 예측력의 적정성을 평가하는 동시에 집단을 구분하는데 가장 기여하는 독립변인들을 파악해 준다(Hair 외, 2006).

판별분석을 활용하기 위해서는 첫째, 최소 사례수는 독립변인의 수보다 많아야 하며, 둘째, 표본평균 분포가 다변량 정규분포를 이루고 있어야 한다. 셋째, 공분산 행렬의 동질성을 가정하고, 넷째, 다중공선성 가정이 가능한 작아야 하며, 마지막으로 독립변인들은 선형적인 관계를 가져야 한다(여운승, 2006). 이 연구의 목적인 소프트웨어 개발자의 전문성 등급과 교육훈련, 자격증, 직무경력에 대한 결정요인의 관계를 고찰하기 위해 다음과 같이 판별분석을 위한 여러 연구모형을 구안하였다.

〈표 III-13〉 전문성 등급을 구분하는 판별모형

모형	판별 함수
노임단가 기준 변인	전문성 등급(G) = f (최종학력, 보유 자격증의 최고 수준, 종사년수)
교육훈련, 자격 및 직무경력 변인	전문성 등급(G) = f (학교교육, 평생교육, 자격증 소지 자격, 프로젝트 수행 자격, 직무경력)

이 연구에서 수행할 판별분석의 절차는 다음과 같다.

첫째, 판별모형의 적합도를 확인한다. 각 집단 간 혹은 집단내의 공분산 행렬의 동일성을 확인하기 위해 Box' M 값의 유의도를 토대로 적합도를 확인한다. 또한 독립변인들에 걸쳐 집단 간에 차이가 있는지를 검증하기 위해 윌크스람다(Wilk' s Lamda) 값의 유의도를 확인하여 등급분류 집단 간 판별점수의 차이를 살펴본다.

둘째, 판별분석 및 효력지수를 확인한다. 모형의 적합도가 검증되었다면 이를 기준으로 판별적재값인 구조행렬값으로 각 모형별 판별력의 중요도 순서를 파악한다. 다시말해, 전문성 등급을 구분함에 있어 가장 판별력이 큰 변인을 파악할 수 있다. 이는 판별함수에 대한 독립변인들의 상대적 기여도를 종합적으로 표현하는 효력지수는 각 함수별 효력지수의 합으로 표현된다(여운승, 2006).

$$\text{효력지수} = \sum(\text{판별적재값}^2 \times \text{상대적고유값})$$

$$* \text{상대적고유값} = \text{각 함수의 고유값} / \sum \text{고유값}$$

셋째, 분류정확률(hit ratio)을 확인한다. 분류정확률이란 분류기준이 된 변인을 활용하여 등급을 분류한 것에 대한 정확도를 의미한다. 이를 통해 어떤 변인을 활용한 경우 분류의 정확도가 높아지는지 확인할 수 있다.

IV. 연구 결과

1. 소프트웨어 개발자의 일반적 특성 및 변인 측정

가. 학교교육

소프트웨어 개발자의 학교교육은 기초조사를 토대로 입직 시 학력, 현재 학력, 입직 시 학교급별 소프트웨어 개발 관련 전공, 현재 학교급별 소프트웨어 개발 관련 전공을 분석하였으며, 학력과 전공으로 학교교육 점수를 계산하였다.

1) 입직 시 학력

응답한 소프트웨어 개발자의 입직 시 학력을 살펴보면, 4년제 대학 졸업이 360명(70.4%), 석사졸업 63명(12.3%), 2년제 전문대 졸업 53명(10.4%)의 순으로 높게 나타났다. 전문대(2년제) 중퇴자는 1명, 3년제 중퇴자는 2명, 4년제 대학 중퇴자는 6명이었으며, 석사과정을 수료하거나 재학중인 학생은 6명, 박사과정에 재학 중이거나 수료자는 3명으로 확인되었다.

〈표 IV-1〉 응답한 소프트웨어 개발자의 입직 시 학력

구분		입직 시	
		빈도(명)	백분율(%)
고등학교	졸업	22	4.3
전문대 (2년제)	중퇴	1	0.2
	졸업	52	10.2
전문대 (3년제)	중퇴	2	0.4
	졸업	1	0.2
대학	중퇴	3	0.6
	졸업	373	73.4

〈계속〉

구분		입직 시	
		빈도(명)	백분율(%)
석사	재학/수료	2	0.4
	졸업	50	9.8
박사	재학/수료	1	0.2
	졸업	1	0.2
소계		508	100.0

주) 결측치(N=26) 제외

2) 현재 최종 학력

응답한 소프트웨어 개발자의 현재 최종학력을 살펴보면, 4년제 대학 졸업이 361명 (70.6%), 석사졸업 68명(13.3%), 2년제 전문대 졸업 42명(8.2%)의 순으로 높게 나타났다. 전문대(2년제) 중퇴자는 1명, 3년제 중퇴자는 2명, 4년제 대학 중퇴자는 6명이었으며, 석사과정을 수료하거나 재학중인 학생은 5명, 박사과정에 재학 중이거나 수료자는 3명으로 확인되었다.

〈표 IV-2〉 응답한 소프트웨어 개발자의 현재 최종학력

구분		빈도(명)	백분율(%)
전문대 (2년제)	중퇴	1	0.2
	졸업	42	8.2
전문대 (3년제)	중퇴	2	0.4
	졸업	14	2.7
대학	중퇴	6	1.2
	졸업	361	70.6
석사	재학/수료	5	1.0
	졸업	68	13.3
박사	재학/수료	3	0.6
	졸업	9	1.8
소계		511	100.0

주) 결측치(N=15) 제외

3) 입직 시 소프트웨어 개발 전공 여부

응답한 소프트웨어 개발자의 입직 시 전공을 살펴보면, 소프트웨어 관련 전공자의 비율은 70.5%였으며, 비전공자의 비율은 29.5%로 나타났다.

〈표 IV-3〉 응답한 소프트웨어 개발자의 입직 시 학교급별 전공

구분	계	소프트웨어 관련		비관련	
		빈도(명)	백분율 (%)	빈도(명)	백분율 (%)
고등학교	25	6	1.8	19	11.0
전문대(2년제)	52	33	9.8	19	11.0
전문대(3년제)	1	1	0.3	0	0.0
4년제	382	260	77.4	122	70.5
석사	50	35	10.4	15	8.7
박사	2	1	0.3	1	0.6
계	512	336	100.0	173	100.0

주 1) 중퇴, 재학 및 수료인원은 해당 학교급에 포함함

주 2) 결측치(N=6) 제외

4) 현재 최종 소프트웨어 개발 전공 여부

응답한 소프트웨어 개발자의 전공 여부를 분석한 결과, 대학교와 대학원 모두 포함하여 평균 67.5%가 소프트웨어 개발 관련 전공을 졸업한 것으로 나타났다. 그 중 박사과정에서의 소프트웨어 개발 전공 비율은 100.0%(5명)으로 가장 높았으며, 석사과정 75.6%(62명), 전문대학 71.4%(5명), 4년제 대학 75.6%(62명)의 순으로 나타났다.

〈표 IV-4〉 응답한 소프트웨어 개발자의 현재 학교급별 전공

구분	계	소프트웨어 관련		비관련	
		빈도(명)	백분율 (%)	빈도(명)	백분율 (%)
전문대(2년제)	44	23	52.3	21	47.7
전문대(3년제)	7	5	71.4	2	28.6
4년제	361	242	67.0	119	33.0
석사	82	62	75.6	20	24.4
박사	5	5	100.0	0	0.0
계	499	337	67.5	162	32.5

주1) 중퇴, 재학 및 수료인원은 해당 학교급에 포함함

주2) 결측치 제외

5) 학교교육 변수

입직 시 학력과 학교급별 소프트웨어 개발 관련 전공 여부를 토대로 도출한 학교교육 [공식 1] (p.80)으로 구한 소프트웨어 개발자의 학교교육 점수는 <표 IV-5>와 같다. 소프트웨어 개발자 학교교육의 평균은 3.56으로 4년제 대학과 석사의 중간 정도로 확인되었다. 학교교육 점수의 범위는 고등학교 비전공으로 졸업한 경우 최소값 1부터 박사를 졸업하고 모든 학교급에서 소프트웨어를 전공한 경우 최대값 9까지의 범위이며, 조사결과 최소값 1, 최대값 6.625로 나타났다.

<표 IV-5> 소프트웨어 개발자의 학교교육 점수

변인명	빈도(명)	평균	표준편차	최소값	최대값	범위
학교교육 점수	516	3.56	1.117	1	6.625	1~9

주1) 결측치(N=10) 제외

나. 평생교육

소프트웨어 개발자의 평생교육은 기초조사를 토대로 입직 후 상급학교로의 진학현황, 입직 후 진학한 상급학교에서의 전공을 분석하였으며, 입직 후 개인의 자발적인 의지에 따라 참여한 훈련 및 입직 후 상급학교의 진학과 훈련으로 계산한 평생교육 점수를 계산하였다.

1) 입직 후 상급학교 진학

응답한 소프트웨어 개발자의 입직 후 상급학교에 진학한 학력을 분석한 결과, 47명이 상급학교에 진학한 것으로 나타났다. 4년제 대학에서 대학원에 진학한 비율이 2, 3년제 전문대학에서 4년제 대학으로 진학한 비율이 21.3%(10명), 고등학교에서 전문대학에 진학한 비율이 8.5%(4명)으로 나타났다.

〈표 IV-6〉 입직 시와 현재 대비 학력 변화 빈도

입직 시 학력	현재 학력	빈도(명)	백분율(%)
고등학교	전문대(2, 3년) 졸업	4	8.5
	대학졸업 또는 중퇴	2	4.3
	석사졸업 또는 재학	3	6.4
전문대(2, 3년)	대학(4년제)	10	21.3
	석사재학	1	2.1
대학(4년제)	석사졸업 또는 재학	23	48.9
	박사졸업 또는 재학	4	8.5
소 계		47	100.0

2) 입직 후 진학한 상급학교에서의 전공

응답한 소프트웨어 개발자의 입직 후 상급학교에 진학한 학력의 전공을 분석한 결과, 47명 중 소프트웨어 관련 전공자는 36명(76.6%)이었으며, 소프트웨어 관련이 아닌 비관련 전공자는 11명(23.4%)인 것으로 나타났다. 이 상급학교에 진학한 것으로 나타났다.

〈표 IV-7〉 응답한 소프트웨어 개발자의 입직 후 진학 학교에서의 전공

구분	계	소프트웨어 관련		비관련	
		빈도(명)	백분율 (%)	빈도(명)	백분율 (%)
전문대(2, 3년제)	5	4	8.5	1	2.1
4년제	16	13	27.7	3	6.4
석사	25	18	38.3	7	14.9
박사	1	1	2.1	0	0.0
계	47	36	76.6	11	23.4

주) 중퇴, 재학 및 수료인원은 해당 학교급에 포함함

3) 훈련 프로그램 참여

훈련 프로그램 기준으로 분석한 결과, 소프트웨어 관련 훈련 프로그램은 평균 4.24개 이수하였으며, 훈련 프로그램 시간으로는 374.39시간 이수한 것으로 나타났다. 이를 프로그램 1개당 훈련시간으로 환산한 결과로 개당 88.30시간으로 평균 10일~15일 정도의 2주 과정의 훈련임을 알 수 있었다. 소프트웨어 비관련 훈련 프로그램 기준으로 분석한 결과, 소프트웨어 비관련 훈련 프로그램은 평균 12.69개 이수하였으며, 훈련 프로그램 시간으로는 288.29시간 이수한 것으로 나타났다. 이를 프로그램 1개당 훈련시간으로 환산하면, 개당 22.72시간으로 평균 2박 3일, 3박 4일짜리 훈련임을 알 수 있었다.

〈표 IV-8〉 소프트웨어 관련 훈련 개수 및 시간(결측치 포함)

구분	소프트웨어 관련 훈련			비관련 훈련		
	빈도	평균	표준편차	빈도	평균	표준편차
훈련프로그램 개수	429	4.24	10.495	406	12.69	61.191
훈련프로그램 시간	420	374.39	1,412.229	398	288.29	1,987.947
프로그램 1개당 훈련시간	88.30			22.72		

주) 훈련시간에 0으로 응답한 경우 값 그대로 반영함

훈련시간에 0시간으로 응답한 경우 실제 훈련에 참여한 적인 없는 경우이므로 훈련에 참여한 사람만을 대상으로 소프트웨어 관련 훈련 개수 및 시간을 분석하였다. 소프트웨어 관련 훈련 프로그램은 평균 5.69개 이수하였으며, 훈련 프로그램 시간으로는 503.02시간 이수한 것으로 나타났다. 이를 프로그램 1개당 훈련시간으로 환산한 결과로 개당 88.40시간으로 평균 10일~15일 정도의 2주 과정의 훈련임을 알 수 있었다. 소프트웨어 비관련 훈련 프로그램 기준으로 분석한 결과, 소프트웨어 비관련 훈련 프로그램은 평균 19.13개 이수하였으며, 훈련 프로그램 시간으로는 444.14시간 이수한 것으로 나타났다.

〈표 IV-9〉 소프트웨어 관련 훈련 개수 및 시간(결측치 제외)

구분	소프트웨어 관련 훈련			비관련 훈련		
	빈도	평균	표준편차	빈도	평균	표준편차
훈련프로그램 개수	318	5.69	11.843	269	19.13	74.397
훈련프로그램 시간	312	503.02	1,619.300	258	444.14	2,456.690
프로그램 1개당 훈련시간	88.40			23.21		

주) 훈련시간에 0으로 응답한 경우 결측치로 처리함

4) 입직 시 학력별 소프트웨어 훈련 프로그램 참여 수준

정규교육과 평생교육의 관계를 살펴보기 위해, 학력 및 상급학교 진학 여부를 분석하였다. 상급학교에 진학하지 않은 집단의 소프트웨어 관련 훈련 프로그램 이수 시간은 평균 306시간이었으나, 상급학교에 진학한 집단의 이수시간은 688시간으로 2배 이상의 차이를 나타냈다.

〈표 IV-10〉 입직 시 학력별 소프트웨어 훈련 프로그램 참여 정도

학교급 유형	빈도(명)	소프트웨어 관련 훈련 개수	소프트웨어 관련 훈련 시간
전문대(2년제)	54	4.54	285.81
전문대(3년제)	17	1.5	120
대학	366	3.79	345.85
석사	69	7.74	410.15
박사	5	0	0
소계	511	4.37	347.55

주 1) 결측치(N=15) 제외

주 2) 위 학교급은 졸업 기준임.

5) 평생교육 변수

앞서 도출한 평생교육 함수 [공식 2] (p.84)를 통해 구한 소프트웨어 개발자의 평생교육 수준은 <표 IV-11>과 같다. 평생교육의 평균은 0.5이었으며, 훈련에 단 1시간이라도 참여한 경험이 있는 집단의 경우에는 평균이 0.7 수준으로 확인되었다. 소프트웨어 개발자의 평생교육은 입직 이후 상급학교에 진학하지 않고, 소프트웨어 관련 훈련과 비관련 훈련 모두 참여하지 않은 경우 최소값 0부터 입직 이후 상급학교에 진학한 경우가 얻을 수 있을 최고 점수 8점과 소프트웨어 관련 훈련과 비관련 훈련 모두 최상위 참여집단일 경우 최대값 9를 갖는다.

<표 IV-11> 소프트웨어 개발자의 평생교육 점수

변인명	빈도(명)	평균	표준편차	최소값	최대값	범위
평생교육 점수	526	0.5	0.48	0	3.9	0-9

평생교육 점수는 학교급에 따라 구분해서 해석될 수 있다. 평생교육의 경우 정규교육에 비해 수준을 구분하기 어려우나 대체적으로 정규교육의 학력 수준을 기준으로 그보다 비슷하거나 높은 수준의 훈련에 참여하므로 학교급별 평생교육의 수준을 분석하였다. 고등학교 졸업자의 경우 평균보다 낮은 0.31수준으로 참여하였으며, 석사와 박사급은 0.8 이상의 높은 평생교육 참여를 보였다.

<표 IV-12> 입직 시 학교급별 평생교육 점수

입직시 학교급	학교급별 점수	평생교육 점수 평균	표준편차	최소값	최대값
고등학교	1	.31	0.330	0	1
전문대(2,3년제)	2	.49	0.462	0	2.35
대학	3	.44	0.383	0	2.70
석사	4	.85	0.716	0	3.90
박사	5	.80	1.216	0	2.20

다. 자격증 소지 자격

소프트웨어 개발자의 자격증 소지 자격은 기초조사를 토대로 보유한 자격의 최고 수준, 자격의 중복 보유 현황, 자격 수준별 보유 개수를 분석하였으며, 보유한 자격증의 수준, 유형, 개수로 자격증 소지 자격 점수를 계산하였다.

1) 보유한 자격의 최고 수준

응답한 소프트웨어 개발자가 보유한 자격증 최고 수준의 자격에 대한 분석을 실시하였다. 그 결과 기사가 340명(64.6%)으로 가장 많았으며, 미보유한 사람이 137명(26.0%)로 그 다음으로 높은 순으로 나타났다. 그 외 기술자 18명(3.4%), 산업기사 20명(3.8%), 기능사 11명(2.1%)로 확인되었다.

〈표 IV-13〉 응답한 소프트웨어 개발자의 최고 자격증 수준

구분	빈도(명)	백분율(%)
기술사	18	3.4
기사	340	64.6
산업기사	20	3.8
기능사	11	2.1
미보유	137	26.0
계	526	100.0

주1) 국가공인민간자격과 해외자격은 국가기술자격 중 기사 수준에 해당됨(한국산업인력공단, 2014)

2) 자격의 중복 보유 유형

응답한 소프트웨어 개발자가 보유한 자격의 중복 보유 현황을 살펴본 결과, 기사자격만을 소지한 경우가 243명(62.5%)로 가장 높았으며, 기사자격과 기능사자격을 보유한 경우가 50명(12.9%), 기사와 산업기사 자격을 소지한 경우가 33명(8.5%), 산업기사 자격만을 소지한 경우가 16명(4.1%) 등의 순으로 높게 나타났다.

〈표 IV-14〉 소프트웨어 개발 자격 중복 유형별 응답한 개발자 비율

자격증 중복 유형				빈도(명)	백분율(%)
기술사	기사			10	2.6
기술사	기사	산업기사		4	1.0
기술사	기사	산업기사	기능사	4	1.0
	기사			243	62.5
	기사	산업기사		33	8.5
	기사	산업기사	기능사	14	3.6
	기사		기능사	50	12.9
		산업기사		16	4.1
		산업기사	기능사	4	1.0
			기능사	11	2.8
계				389	100.0

주1) N=389

주2) 기사자격에는 국가공인민간자격, 해외자격도 포함됨.

3) 자격 수준별 보유 개수

응답한 소프트웨어 개발자가 보유한 자격 수준별 자격증의 개수를 살펴 본 결과, 대체적으로 자격수준별 1개를 소지한 비율이 높았다. 기술사의 경우 1개 소지자 11명 (61.1%), 기사자격 281명 (92.7%), 국가공인민간자격 46명 (62.2%), 해외자격 66명 (56.9%), 산업기사 70명 (93.3%), 기능사 54명 (62.8%)로 나타나 대부분 1개를 소지하고 있음을 알 수 있었다.

〈표 IV-15〉 응답한 소프트웨어 개발자의 수준별 소지 자격 수 빈도

구분	계 (개)	계 (명)	1개		2개		3개 이상	
			빈도(명)	백분율(%)	빈도(명)	백분율(%)	빈도(명)	백분율(%)
기술사	26	18	11	61.1	6	33.3	1	5.6
기사	국가기술자격	337	281	92.7	19	6.3	3	1.0
	공인민간자격	115	46	62.2	19	25.7	9	12.2
	해외자격	190	66	56.9	38	32.8	12	10.3
산업기사	80	75	70	93.3	5	6.7	-	-
기능사	123	86	54	62.8	27	31.4	5	5.8

주1) 중복소지

주2) 국가공인민간자격과 해외자격은 국가기술자격 중 기사 수준에 해당됨(한국산업인력공단, 2014)

4) 자격증 소지 자격 변수

개인이 소지하고 있는 자격증 중 최고 수준의 자격증과 같은 등급 내에서의 자격증 보유 개수, 자격보유유형 등을 고려해서 도출한 자격증 소지 자격의 합수[공식 3] (p.87)를 통한 소프트웨어 개발자의 자격증 소지 자격 수준은 〈표 IV-16〉와 같다. 자격증 소지 자격의 평균은 2.37로 산업기사 수준이나, 자격증을 소지 않은 0의 경우를 모두 포함하고 있어, 낮은 수준이었으나 자격증을 소지하고 있는 경우로 한정된 경우에는 평균은 3.16으로 기사수준으로 확인되었다. 소프트웨어 개발자의 자격증 소지 자격은 자격증을 전혀 소지하지 않은 경우 최소값 9부터 소프트웨어 관련 기술자 자격 6개를 모두 소지하고 있으며 관련된 기술사 하위 자격인 기사, 산업기사, 기능사자격을 모두 소지한 경우 최대값 5를 갖는다.

〈표 IV-16〉 소프트웨어 개발자의 자격증 소지 자격 점수

변인명	빈도(명)	평균	표준편차	최소값	최대값	범위
자격증 소지 자격 점수	526	2.37	1.53	0	4.63	0~5

주) 자격증을 소지하지 않은 경우 '0' 으로 두었기 때문에 결측치 없음

라. 프로젝트 수행 자격

소프트웨어 개발자의 프로젝트 수행 자격은 기초조사를 토대로 직급별 참여한 프로젝트 개수를 분석하였으며, 이를 토대로 프로젝트 수행 참여 점수를 계산하였다.

1) 직급별 프로젝트 참여 개수

응답한 소프트웨어 개발자의 직급별 프로젝트 참여를 살펴보면, 평균 15.15개의 프로젝트에 참여한 것으로 나타났다. 과차장급에서의 프로젝트 참여 수가 9.08개로 가장 높았으며, 팀부장급 7.44개, 임원급 7.00개, 사원급 6.66개, 대리급 5.85개의 순으로 나타났다.

〈표 IV-17〉 응답한 소프트웨어 개발자가 직급별 프로젝트 참여 개수

구분	빈도(명)	평균(개)	표준편차	최소값	최대값
사원급	425	6.66	24.897	1	500
대리급	326	5.85	7.834	1	100
과차장급	171	9.08	13.756	1	100
팀부장급	27	7.44	5.807	1	20
임원급	8	7.00	6.866	1	20
총 개수	-	15.15	35.660	1	650

2) 프로젝트 수행 자격 점수

직급별로 참여한 프로젝트 수행 개수를 반영하여 프로젝트 수행 자격 함수 [공식 4] (p.91)를 구성하였으며, 이 공식을 통해 도출한 소프트웨어 개발자 프로젝트 수행 자격 수준은 〈표 IV-18〉과 같았다. 프로젝트 수행 자격의 평균은 1.29로 사원에서 대리급의 중간 정도 수준으로 확인되었으며, 이 점수는 프로젝트에 전혀 참여해보지 못한 경우 최소값 0부터 임원급까지 도달하면서 모든 급에서 최고 수준(75%이상)의 프로젝트 참여 수를 보유한 경우 최대값 5의 범위에서의 점수를 의미한다.

〈표 IV-18〉 소프트웨어 개발자의 프로젝트 수행 자격 점수

변인명	빈도(명)	평균	표준편차	최소값	최대값	범위
프로젝트 수행 자격 점수	526	1.29	0.841	0.25	4.25	0~5

마. 직무경력

소프트웨어 개발자의 직무경력에 기초조사를 토대로 총 경력기간, 직급별 경력기간, 분야별 경력기간을 분석하였으며, 개인이 도달한 최고 수준의 직급과 소프트웨어 개발 재직년수로 직무경력 점수를 계산하였다.

1) 총 경력기간

응답한 소프트웨어 개발자의 총 경력기간을 분석한 결과, 평균 9.16년 종사한 것으로 확인되었으며, 소프트웨어 관련 경력기간은 7.70년 소프트웨어 관련 외 경력기간은 4.47년으로 나타났다.

〈표 IV-19〉 응답한 소프트웨어 개발자의 총 경력기간

구분	빈도(명)	평균(년)	표준편차	최소값	최대값
총 경력기간	455	9.16	7.181	1	50
소프트웨어 종사년수	448	7.70	5.964	1	39
소프트웨어 외 종사년수	160	4.47	4.795	1	25

주) 결측치에 차이가 있어 소프트웨어 종사년수와 소프트웨어 외 종사년수의 합이 총 경력기간 평균과 상이함

2) 직급별 경력기간

응답한 소프트웨어 개발자의 직급별 경력기간을 분석한 결과, 임원급에서의 경력이 4.67년으로 가장 높았으며, 과차장급 4.81년, 팀부장급 4.32년, 사원급 3.98년, 대리급 3.67년으로 나타났다.

〈표 IV-20〉 응답한 소프트웨어 개발자의 직급별 경력기간

구분	빈도(명)	평균(년)	표준편차	최소값	최대값
사원급	455	3.98	3.086	0	23
대리급	354	3.67	2.485	0	26
과차장급	178	4.81	3.293	0	18
팀부장급	37	4.32	3.823	0	20
임원급	9	4.67	3.428	1	10

3) 분야별 경력기간

응답한 소프트웨어 개발자의 분야별 경력기간을 살펴보면, 소프트웨어 분야에 종사하 되, 사원에서 과차장급까지는 소프트웨어 개발 관련 종사기간이 비관련 기간에 비해 높은 것으로 나타났으나, 팀부장급으로 갈수록 소프트웨어 개발 비관련 직무에 종사하는 것으로 나타났다. 따라서, 팀부장급 이상에서는 직접적인 개발보다는 관리 등의 직무를 수행함을 알 수 있었다.

〈표 IV-21〉 응답한 소프트웨어 개발자의 분야별 경력기간

구분	소프트웨어 개발 관련 경력기간			비관련 경력기간		
	빈도(명)	평균(년)	표준편차	빈도(명)	평균(년)	표준편차
사원급	432	3.51	2.745	121	2.44	1.875
대리급	334	3.28	1.652	72	2.83	2.162
과차장급	166	4.26	2.830	43	3.35	2.497
팀부장급	29	3.10	1.448	18	3.33	3.308
임원급	8	3.75	3.012	3	4.00	2.646

4) 직무경력 변수

지금까지 도달한 최고 수준의 직급과 직급별 종사년수를 반영하여 직무경력 함수 [공식 5] (p.93)를 구성하였으며, 이 공식을 통해 도출한 소프트웨어 개발자의 직무경력 수준은 <표 IV-22>과 같다. 직무경력의 평균은 3.68로 확인되었다. 소프트웨어 개발자의 직무경력에 입사하여 사원인 경우 최소값 2부터 임원급에서 소프트웨어 관련 재직년수의 중위수 기준으로 재직년수 최상위집단인 경우 최대값 9를 갖는다.

<표 IV-22> 소프트웨어 개발자의 직무경력 점수

변인명	빈도(명)	평균	표준편차	최소값	최대값	범위
직무경력 점수	526	3.68	1.508	1.5	7.5	1.5~7.5

2. 소프트웨어 개발자의 전문성 발달 수준과 등급

가. 전문성 발달 수준 도구 문항별 반응 분포

응답한 소프트웨어 개발자의 전문성 발달 수준을 측정하는 각 문항별 반응 분포를 제시하였다. 척도의 각 문항은 전문성 발달 수준을 의미하는 1수준부터 5수준까지에 해당되는 각 단계에 대한 개인의 동의 정도를 조사하고 있다. 소프트웨어 개발자의 전문성 발달 수준을 측정하는 문항은 응답자의 동의 정도에 따라 5점 척도(매우 잘 못함(1): 지식과 기술을 배워 이론적으로 알고는 있지만 다른 동료들에 비해 실제 적용하거나 활용하는 업무를 수행하기에는 어렵다고 평가되는 수준, 잘 못함(2): 지식과 기술을 적용하거나 활용할 수 있어 하나의 업무가 주어지지만 다른 동료들에 비해 그 결과에 대해서는 완벽하지 않다고 평가되는 수준, 보통(3): 독립적으로 하나의 업무를 맡고 있으며, 다른 동료들에 비해 업무 결과에 대해서도 신뢰받고, 상황에 맞는 의사결정과 판단이 가능하다고 평가되는 수준, 잘함(4): 나만의 노하우로 직관적인 업무 처리가 가능하며 다른 동료들에 비해 우수한 결과를 보인다고 평가되는 수준, 매우 잘함(5): 다른 동료들에

비해 판단과 결정이 무의식적으로 자동화되어 일어나며, 관련 지식과 기술의 통합적 사고가 가능하며 내가 내린 직관적 판단에 대한 비판적 성찰까지 가능하다고 평가되는 수준)에 반응하도록 구성하였으며 문항별 이론적 평균은 3점이다.

분석 및 설계는 소프트웨어 생명주기 단계상 가장 먼저 진행하는 단계로 전반적인 프로젝트의 방향성과 큰 틀을 잡는 단계이다. 소프트웨어 분야에서는 분석 및 설계의 수준이 높아질수록 전문성이 높아진다고 보고되고 있다. 분석 및 설계의 문항별 평균은 2.32~3.45로 보통 수준을 나타내었다. 각 문항별 동의정도를 이론적 평균을 기준으로 구분하면 7, 8, 9번 문항에 대해서는 보통보다 낮은 수준으로 나타났으며 나머지 문항에 대해서는 보통수준으로 나타났다.

구현 단계는 소프트웨어 생명주기 단계상 분석 및 설계 다음에 진행된다. 분석 및 설계를 통해 큰 프레임을 잡았다면 세부적인 내용을 코딩하는 단계가 구현이다. 따라서 소프트웨어 개발자로 입직하여 가장 먼저 수행하는 직무가 구현에 해당된다. 구현의 문항별 평균은 2.69~3.42로 보통 수준을 나타내었다. 각 문항별 동의정도를 이론적 평균을 기준으로 구분하면 8, 9, 10, 11번 문항에 대해서는 보통보다 낮은 수준으로 나타났으며 나머지 문항에 대해서는 보통수준 또는 잘함의 수준으로 나타났다.

테스팅 단계는 소프트웨어 생명주기 단계상 3번째 단계로 구현된 결과물에 대해 에러가 없는지, 요구를 잘 반영했는지 등을 점검하는 단계다. 구현과 전반적인 설계 내용을 숙지한 수준에 도달한 경우 테스팅 업무가 주어진다. 테스팅의 문항별 평균은 2.71~3.23으로 보통 수준을 나타내었다. 각 문항별 동의정도를 이론적 평균을 기준으로 구분하면 10번 문항에 대해서는 보통보다 낮은 수준으로 나타났으며 나머지 문항에 대해서는 보통수준으로 나타났다.

프로젝트 관리는 소프트웨어의 생명주기 단계에 속한 단계는 아니지만, 소프트웨어 개발에 있어 반드시 포함되어야 하는 영역이다. 프로젝트를 정해진 기한과 예산에 맞춰 운영하고, 전반적으로 인력 배치, 시간관리, 품질관리 등의 총체적인 역할을 수행하는 직무로 중급이상의 개발자들이 주로 수행한다. 소프트웨어 분야에서는 분석 및 설계과 더불어 전문가 수준으로 올라갈수록 프로젝트 관리 능력을 요구받는다. 프로젝트 관리의 문항별 평균은 2.32~3.45로 보통 수준을 나타내었다. 각 문항별 동의정도를 이론적 평균을 기준으로 구분하면 2, 6번 문항에 대해서는 보통보다 낮은 수준으로 나타났으며 나머지 문항에 대해서는 보통수준으로 나타났다.

〈표 IV-23〉 소프트웨어 개발자의 전문성 발달 수준 측정도구에 대한 문항별 반응 분포

문항	동의정도					빈도	평균	표준 편차
	매우 못함	잘 못함	보통	잘함	매우 잘함			
[분석 및 설계]								
1. 요구사항 확인 능력	15 (2.9)	53 (10.2)	216 (41.6)	184 (35.5)	51 (9.8)	519 (100.0)	3.39	.902
2. 어플리케이션 설계 능력	19 (3.7)	72 (13.9)	226 (43.6)	154 (29.7)	47 (9.1)	518 (100.0)	3.27	.936
3. 화면 구현 능력	16 (3.1)	53 (10.2)	197 (38.0)	186 (35.8)	67 (12.9)	519 (100.0)	3.45	.947
4. 데이터 베이스 요구사항 분석 능력	19 (3.7)	75 (14.5)	200 (38.8)	173 (33.5)	49 (9.5)	516 (100.0)	3.31	.957
5. 하드웨어 분석 능력	53 (10.2)	153 (29.5)	205 (39.5)	85 (16.4)	23 (4.4)	519 (100.0)	2.75	.993
6. 개념 데이터 모델링 능력	30 (5.8)	95 (18.5)	215 (41.8)	137 (26.7)	37 (7.2)	514 (100.0)	3.11	.981
7. 펌웨어 분석 설계 능력	123 (23.8)	161 (31.2)	154 (29.8)	61 (11.8)	17 (3.3)	516 (100.0)	2.40	1.073
8. 임베디드 애플리케이션 분석 설계 능력	143 (27.7)	129 (25.0)	144 (27.9)	73 (14.1)	28 (5.4)	517 (100.0)	2.45	1.188
9. 디바이스 드라이버 분석 설계 능력	150 (29.1)	142 (27.5)	151 (29.3)	56 (10.9)	17 (3.3)	516 (100.0)	2.32	1.102
10. 소프트웨어 아키텍처 설계 능력	45 (8.7)	121 (23.4)	202 (39.1)	118 (22.8)	31 (6.0)	517 (100.0)	2.94	1.024
11. 논리 데이터베이스 설계 능력	40 (7.8)	89 (17.2)	211 (65.9)	142 (27.5)	34 (6.6)	516 (100.0)	3.08	1.008
12. 데이터 품질관리 능력	43 (8.3)	107 (29.0)	208 (40.2)	124 (24.0)	35 (6.8)	517 (100.0)	3.00	1.026
13. 데이터 표준화 능력	42 (8.1)	99 (19.1)	217 (42.0)	125 (24.2)	34 (6.6)	517 (100.0)	3.02	1.011

〈계속〉

문항	동의정도					빈도	평균	표준 편차
	매우 못함	잘 못함	보통	잘함	매우 잘함			
[구현]								
14. 소프트웨어 아키텍처 이 행 능력	20 (3.8)	63 (12.1)	220 (43.5)	177 (34.0)	40 (7.7)	520 (100.0)	3.30	.915
15. SQL 활용 능력	29 (5.6)	47 (9.1)	199 (38.4)	179 (34.6)	64 (12.4)	518 (100.0)	3.39	1.002
16. 소프트웨어 공학 활용 능력	25 (4.8)	87 (16.7)	243 (46.7)	140 (26.9)	25 (4.8)	519 (100.0)	3.10	.901
17. 어플리케이션 구현 능력	17 (3.3)	62 (12.0)	225 (43.6)	155 (30.0)	57 (11.0)	516 (100.0)	3.34	.940
18. 데이터 입출력 구현 능력	22 (4.3)	75 (18.8)	232 (44.9)	136 (26.3)	52 (10.1)	517 (100.0)	3.23	.963
19. 제품소프트웨어 패키징 능력	45 (8.7)	111 (21.3)	221 (42.5)	116 (22.3)	27 (5.2)	514 (100.0)	2.94	.994
20. 통합구현 능력	33 (6.4)	99 (19.1)	224 (43.2)	128 (24.7)	34 (6.6)	518 (100.0)	3.06	.977
21. 펌웨어 구현 능력	134 (25.8)	159 (30.6)	149 (28.7)	59 (11.4)	18 (3.5)	517 (100.0)	2.36	1.088
22. 운영체제 이식 능력	157 (30.3)	142 (27.4)	152 (29.3)	49 (9.4)	19 (3.7)	516 (100.0)	2.29	1.105
23. 디바이스 드라이버 구현 능력	135 (26.0)	142 (27.4)	159 (30.6)	61 (11.8)	22 (4.2)	517 (100.0)	2.41	1.120
24. 임베디드 애플리케이션 구현 능력	131 (25.3)	134 (25.9)	170 (32.8)	62 (12.0)	21 (4.1)	518 (100.0)	2.44	1.112
25. 네트워크 프로그래밍 구 현 능력	72 (13.8)	123 (23.7)	184 (35.4)	118 (22.7)	23 (4.4)	520 (100.0)	2.80	1.076
26. 기술문서 개발 능력	17 (3.3)	76 (14.7)	226 (43.6)	149 (28.8)	50 (9.7)	518 (100.0)	3.27	.939
[테스팅]								
27. 테스트 계획 능력	14 (2.7)	58 (11.1)	244 (46.8)	154 (29.6)	51 (9.8)	521 (100.0)	3.33	.895
28. 테스트 환경 구축 능력	14 (2.7)	69 (13.3)	237 (45.6)	163 (31.3)	37 (7.1)	520 (100.0)	3.27	.876
29. 개발자 테스트 능력	15 (2.9)	42 (8.1)	225 (43.4)	180 (34.7)	56 (10.8)	518 (100.0)	3.42	.893
30. 테스트 분석 능력	20 (3.9)	64 (12.4)	219 (43.5)	172 (33.2)	43 (8.3)	518 (100.0)	3.30	.925
31. 테스트 설계 능력	24 (4.6)	64 (12.3)	227 (43.7)	157 (30.2)	48 (9.2)	520 (100.0)	3.27	.952

<계속>

문항	동의정도					빈도	평균	표준 편차
	매우 못함	잘 못함	보통	잘함	매우 잘함			
32. 테스트 실행 능력	19 (3.7)	49 (9.4)	206 (39.7)	196 (37.8)	49 (9.4)	519 (100.0)	3.40	.916
33. 테스트 자동화 능력	33 (6.3)	128 (24.6)	240 (46.2)	88 (16.9)	31 (6.0)	519 (100.0)	2.92	.950
34. 테스트 관리 능력	27 (5.2)	74 (14.3)	224 (43.2)	153 (29.5)	41 (7.9)	519 (100.0)	3.21	.960
35. 테스트 완료 능력	19 (3.7)	59 (11.3)	225 (43.3)	165 (31.7)	52 (10.0)	520 (100.0)	3.33	.932
36. UI 테스트 능력	21 (4.0)	74 (14.3)	196 (37.8)	174 (33.5)	54 (10.4)	519 (100.0)	3.32	.977
[프로젝트 관리]								
37. IT 비즈니스 환경 분석 능력	46 (8.8)	118 (22.7)	219 (42.1)	108 (20.8)	29 (5.6)	520 (100.0)	2.92	1.003
38. ITSM(IT Service Management) 시스템 활용 능력	73 (14.0)	128 (24.6)	217 (41.7)	79 (15.2)	23 (4.4)	520 (100.0)	2.71	1.028
39. IT 비즈니스 커뮤니케이션 능력	27 (5.2)	78 (15.0)	210 (40.4)	161 (31.0)	44 (8.5)	520 (100.0)	3.23	.978
40. IT 기술 관련 문제해결능력	25 (4.8)	89 (17.1)	217 (41.7)	147 (28.3)	42 (8.1)	520 (100.0)	3.18	.969
41. 프로젝트 관리 능력	35 (6.7)	108 (20.7)	215 (41.3)	132 (25.3)	31 (6.0)	521 (100.0)	3.03	.984
42. 프로젝트 성과 평가 능력	48 (9.2)	119 (22.8)	222 (42.5)	105 (20.1)	28 (5.4)	522 (100.0)	2.90	1.001

주) 결측치 제외

나. 전문성 발달 수준 측정

1) 도구에 응답한 전문성 발달

소프트웨어 개발자의 전문성 발달 수준을 살펴보기 위해 전문성 발달 수준의 구성요소인 분석 및 설계, 구현, 테스트, 프로젝트 관리의 응답 평균, 표준편차, 빈도 등의 기술통계를 분석하였다. 4개의 구인은 전문성 발달 수준을 구성함에 있어 문항수에 차이가 났기 때문에 평균으로 변환하여 5점 만점으로 계산하였다. 소프트웨어 개발자의 전문성 발달 수준은 평균 3.02점 수준이었다.

〈표 IV-24〉 전문성 발달 수준 측정도구에 응답한 값

구분	평균	표준편차	최소값	최대값	왜도	첨도
전문성 발달수준	3.02	.626	1.00	5.00	-.330	.838
분석 및 설계	2.94	.706	.38	5.00	-.273	.682
구현	2.91	.683	1.00	5.00	-.084	.359
테스팅	3.26	.763	0.80	5.00	-.365	.830
프로젝트 관리	2.99	.830	.83	5.00	-.208	.234

2) 가중치를 반영한 전문성 발달 수준

전문성 발달 수준을 구성하고 있는 4개 구인의 상관관계를 분석한 결과, .67~.80으로 높은 상관이 나타났으며 .05수준에 모두 유의하였다. 베리맥스 회전방법으로 주성분 분석을 실시한 전문성 발달의 4가지 요소별 가중치는 분석 및 설계(0.384), 구현(0.284), 테스팅(0.286), 프로젝트 관리(0.317)였다. 위 가중치의 합이 1을 넘기 때문에 1을 기준으로 비율을 변경하였다. 그 결과, 분석 및 설계(0.303), 구현(0.224), 테스팅(0.223), 프로젝트 관리(0.250)의 가중치로 확인되었다.

〈표 IV-25〉 전문성 발달 구성요소의 주성분 분석 결과

구성요소	성분점수 계수행렬값	1 변환값
분석 및 설계	0.384	0.303
구현	0.284	0.224
테스팅	0.286	0.223
프로젝트 관리	0.317	0.250

$$S/ WDevelopers' Expertise = 0.303Anal + 0.224Prog + 0.223Test + 0.250Manage$$

Expertise = 소프트웨어 개발자 전문성 발달 수준

Ana = Analysis의 줄임말로 분석설계 문항값의 평균

Pro = Programming의 줄임말로 구현 문항값의 평균

Test = Testing의 줄임말로 테스트 문항값의 평균

Manage = Management의 줄임말로 프로젝트 관리 문항값의 평균

위의 함수식을 통해 도출한 소프트웨어 개발자의 전문성 발달 수준은 평균 3.00 표준 편차는 .651으로 확인되었다. 최소값 1.00, 최대값은 5.00이었으며, 왜도와 첨도를 통해 전문성 발달 점수의 정규성을 확인해 본 결과, 정규분포를 이루는 것으로 확인되었다. 앞으로 이 연구에서 전문성 발달 수준이라고 지칭되는 것은 가중치로 보정한 값을 의미한다.

〈표 IV-26〉 가중치로 보정한 소프트웨어 개발자의 전문성 발달 수준

구분	평균	표준편차	최소값	최대값	왜도	첨도
전문성 발달수준	3.00	.651	0	5.00	.107	1.278
분석 및 설계(.303)	.89	.217	0	1.52	-.369	.986
구현(.224)	.64	.163	0	1.12	-.469	1.528
테스팅(.223)	.73	.178	0	1.12	-.635	1.697
프로젝트 관리(.250)	.74	.214	0	1.25	-.370	.637

주) 분석 및 설계, 구현, 테스트 및 프로젝트 관리를 합한 값이 전문성 발달 수준값임.

다. 소프트웨어 개발자의 전문성 등급 구분

전문성 발달 수준을 구성하는 4개 요소별 K-mean 군집분석을 실시하여 전문성 발달 수준의 등급을 구분하였다.

1) 전문성 발달 수준에 따른 군집분석 결과

K-mean 군집분석을 실시하기에 앞서 Dreyfus & Dreyfus(1980)이 제시한 전문성 5단계 발달 단계를 토대로 군집의 개수를 5개로 지정하여 실시한 결과, 5개로 구분되었다. 가중치가 반영된 함수식에 따른 전문성 발달 수준 평균은 3.02으로 나타났다(<표 IV-27> 참고).

<표 IV-27> 전문성 발달 수준에 따른 군집분석 결과

구분	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
전문성 발달 수준	1.60	2.38	2.97	3.47	4.13
분석설계	.50	.70	.87	1.03	1.25
구현	.36	.53	.64	.75	.87
테스팅	.38	.60	.72	.83	.95
프로젝트 관리	.35	.56	.74	.86	1.05
사례수	30	97	203	147	43
범위	1.00~1.96	1.99~2.67	2.69~3.22	3.23~3.78	3.82~5.00

주 1) 결측치 (N=6) 제외

주 2) 분석 및 설계, 구현, 테스팅 및 프로젝트 관리를 합한 값이 전문성 발달 수준값임.

군집분석 결과가 아닌 5점 척도를 기준으로 전문성 등급을 구분하면 Cluster 5는 평균 4.13으로 다수가 전문가 수준으로 보기에는 어려움이 있으나 4.5이상 전문가 수준에 속하는 표본이 5명 존재하므로 Cluster 5를 Cluster 4와 구분되는 개발 전문가로 명명하고자 한다.

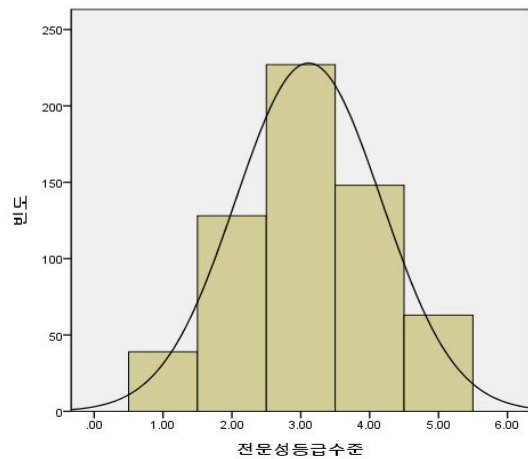
2) 군집분석 결과의 타당화

K-mean 군집분석은 연구자가 임의로 군집의 개수를 지정하기 때문에 군집 개수가 타당한지, 군집이 잘 나누어졌는지에 대한 검토가 필요하다. 일반적으로 동일한 자료에서 다른 변인들을 기준으로 지정된 군집으로 구분했을 때 결과가 유사한지 비교하는 방법과 동일한 자료를 임의로 두 부분으로 구분하여 독립적으로 군집분석을 실시하여 결과를 비교하는 방법이다(김기영 외, 1990). 기존 자료를 짝수변(Sample A)과 홀수변(Sample B)로 구분하여 5개 군집으로 분류한 결과 기존 분석 결과와 유사한 값이 도출되었으며 ANOVA 분석 결과 통계적으로 모두 유의한 결과가 도출되었다. 따라서 5개 군집으로 구분한 결과를 소프트웨어 개발자의 전문성 발달 수준 군집으로 사용해도 적합하다고 볼 수 있다.

〈표 IV-28〉 응답한 소프트웨어 개발자의 전문성 등급의 군집분석 유의도 검사

Cluster 개수	Cluster 구분	Sample	중앙값	F검증
5	1	A	1.31	.000
	2		2.10	
	3		3.15	
	4		3.59	
	5		4.13	
	1	B	1.50	.000
	2		2.39	
	3		3.01	
	4		3.48	
	5		4.10	

전문성 발달 수준 군집별 중심값을 기준으로 등급화한 결과의 정규성을 검증하면 다음과 같다.



[그림 IV-1] 군집별 빈도에 대한 정규성 검증

3) 전문성 발달 수준의 등급화 및 명명화

군집분석 결과로 5개의 집단이 도출되었다. Dreyfus & Dreyfus(1980)이 제시한 전문성 발달과정 5단계를 기초로 조사도구를 구성하여 소프트웨어 개발자가 갖춰야하는 전문성 구성요소를 5단계 리커트 척도로 측정하였다. 전문성 발달 수준은 Dreyfus & Dreyfus(1980)의 결과와 동일하게 5개의 집단으로 구분되며, 이 결과는 통계적으로 나타난 것으로 나타났다.

하지만 군집별 중심점은 cluster 1의 경우 1.60, cluster 2는 2.38, cluster 3은 2.97, cluster 4는 3.47, cluster 5는 4.13으로 나타났다. cluster 1의 경우 1.60으로 1 점대로 가장 낮은 수준인 Cluster 1(초급개발자)로 결정할 수 있으며, cluster 2의 경우 2.38 수준으로 2점대로 Cluster 1(초급개발자)보다 높은 Cluster 2(중급개발자)로 결정할 수 있다. cluster 3의 경우, 2.97로 3에 매우 근접하므로 Cluster 3(고급개발자)로 결정할 수 있으며, cluster 4는 3.47로 cluster 3과 5의 중간 정도 점수이나 cluster 4에 해당하는 표본수가 147명으로 높으므로 cluster 3과 다른 집단으로 구분해도 무방할 것으로 판단하였다. 이에 따라 cluster 4는 Cluster 4(특급개발자)로 결정하였으나,

cluster 5는 4.13으로 4에 가까운 점수라 Dreyfus & Dreyfus(1980)가 최고 전문가 수준으로 제시한 수준으로 보기에 어려움이 있었다. 따라서 cluster 5도 Cluster 4(특급개발자)로 보았다.

노임단가 등급에서 명명한 초급기술자, 중급기술자, 고급기술자, 특급기술자, 기술사와 Dreyfus & Dreyfus(1980)이 제시한 초보자, 고급입문자, 능숙자, 숙련자, 전문가를 바탕으로 군집분석 결과를 바탕으로 전문성 발달 수준의 평균에 따라 명명하면 다음과 같다. Cluster 1은 Cluster 1(초급개발자), Cluster 2는 Cluster 2(중급개발자), Cluster 3은 Cluster 3(고급개발자), Cluster 4는 Cluster 4(특급개발자) 그리고 가장 높은 수준인 Cluster 5는 Cluster 5(개발전문가)이다.

〈표 IV-29〉 군집분석 결과에 따른 전문성 등급

군집분석 결과	전문성 등급명	전문성 발달 평균값	비고	
			Dreyfus & Dreyfus(1980) 전문성 발달 단계	노임단가 등급
Cluster 1	초급개발자	1.60	초보자	초급기술자
Cluster 2	중급개발자	2.38	고급입문자	중급기술자
Cluster 3	고급개발자	2.97	능숙자	고급기술자
Cluster 4	특급개발자	3.47	숙련자	특급기술자
Cluster 5	개발전문가	4.13	전문가	기술사

3. 소프트웨어 개발자의 전문성 등급별 특성

앞서 전문성 발달 수준의 결과를 토대로 구분한 전문성 등급별 교육훈련, 자격 및 직무경력의 특성을 분석하였다.

가. Cluster 1(초급개발자)의 특성

노임단가 등급으로 본 Cluster 1(초급개발자)의 최빈치는 초급기술사(45.2%)였다. 최빈치를 기준으로 한 상위급은 54.48%였으며, 초급기술자가 가장 낮은 수준의 등급으로 하위급에 해당되는 사람은 없었다. Cluster 1(초급개발자)는 4년제 대학 학위 소지자였으며, 4년제 대학 이상인 석사와 박사 졸업생은 13.2%, 하위급인 고등학교 및 전문대학 졸업생은 78.9%였다. 입직 후 진학한 비율은 5.3%였다. 소프트웨어 관련 훈련에는 평균보다 높은 5.85개, 662.72시간으로 평균보다 많이 참여하였으며, 비관련 훈련에는 평균 9.48개, 53.00시간으로 역시 평균보다 많이 참여하였다.

Cluster 1(초급개발자)가 소지한 자격증의 최빈치는 기사 자격이었으며, 자격을 미소지하고 있는 비율은 46%였다. 기사보다 높은 기술사 자격을 소지하고 있는 사람은 없었으며, 산업기사와 기능사 자격 소지 비율은 45.90%였다. 또한, 프로젝트에 참여한 수는 평균 7개였다. Cluster 1(초급개발자)의 총재직년수는 평균 5.37년이었으며, 소프트웨어 개발 직무에 종사한 기간은 4.69년이었다.

나. Cluster 2(중급개발자)의 특성

노임단가 등급으로 본 Cluster 2(중급개발자)의 최빈치는 중급기술사(39.8%)였다. 최빈치를 기준으로 한 상위급은 32.70%였으며, 하위급은 27.6%였다. Cluster 2(중급개발자)는 4년제 대학 학위 소지자였으며, 4년제 대학 이상인 석사와 박사 졸업생은 14.0%, 하위급인 고등학교 및 전문대학 졸업생은 72.9%였다. 입직 후 진학한 비율은 8.4%였다. 소프트웨어 관련 훈련에는 2.34개, 346.45시간으로 평균보다 적게 참여하였으며, 비관련 훈련에는 평균 3.37개, 81.96시간으로 역시 평균보다 적게 참여하였다.

Cluster 2(중급개발자)가 소지한 자격증의 최빈치는 기사 자격이었으며, 자격을 미소지하고 있는 비율은 31%였다. 기사보다 높은 기술사 자격 소지 비율은 1.90%, 산업기사와 기능사 자격 소지 비율은 39.40%였다. 또한, 프로젝트에 참여한 수는 평균 9.37개였다. Cluster 2(중급개발자)의 총재직년수는 평균 7.40년이었으며, 소프트웨어 개발 직무에 종사한 기간은 6.65년이였다.

다. Cluster 3(고급개발자)의 특성

노임단가 등급으로 본 Cluster 3(고급개발자)의 최빈치는 중급기술사(37.9%)였다. 최빈치를 기준으로 한 상위급은 28.60%였으며, 하위급은 33.60%였다. Cluster 3(고급개발자)는 4년제 대학 학위 소지자였으며, 4년제 대학 이상인 석사와 박사 졸업생은 14.6%, 하위급인 고등학교 및 전문대학 졸업생은 70.8%였다. 입직 후 진학한 비율은 11.8%였다. 소프트웨어 관련 훈련에는 3.26개, 395.19시간으로 평균보다 적게 참여하였으며, 비관련 훈련에는 평균 11.57개, 255.67시간으로 역시 평균보다 적게 참여하였다.

Cluster 3(고급개발자)가 소지한 자격증의 최빈치는 기사 자격이었으며, 자격을 미소지하고 있는 비율은 25.6%였다. 기사보다 높은 기술사 자격 소지 비율은 3.30%, 산업기사와 기능사 자격 소지 비율은 36.60%였다. 또한, 프로젝트에 참여한 수는 평균 8.53개였다. Cluster 3(고급개발자)의 총재직년수는 평균 8.53년이었으며, 소프트웨어 개발 직무에 종사한 기간은 7.13년이였다.

라. Cluster 4(특급개발자)의 특성

노임단가 등급으로 본 Cluster 4(특급개발자)의 최빈치는 중급기술사(32.3%)였다. 최빈치를 기준으로 한 상위급은 18.50%였으며, 하위급은 49.30%였다. Cluster 4(특급개발자)는 4년제 대학 학위 소지자였으며, 4년제 대학 이상인 석사와 박사 졸업생은 12.2%, 하위급인 고등학교 및 전문대학 졸업생은 71.2%였다. 입직 후 진학한 비율은 7.0%였다. 소프트웨어 관련 훈련에는 5.08개로 평균보다 많이 참여하였으나 시간으로 보면 289.10시간으로 평균보다 적게 참여하였다. 비관련 훈련에는 평균 11.57개, 255.67시간으로 평균보다 많이 참여하였다.

Cluster 4(특급개발자)가 소지한 자격증의 최빈치는 기사 자격이었으며, 자격을 미소지하고 있는 비율은 28.5%였다. 기사보다 높은 기술사 자격 소지 비율은 5.10%, 산업 기사와 기능사 자격 소지 비율은 38.70%였다. 또한, 프로젝트에 참여한 수는 평균 18.03개였다. Cluster 4(특급개발자)의 총재직년수는 평균 10.71년이었으며, 소프트웨어 개발 직무에 종사한 기간은 9.07년이었다.

마. Cluster 5(개발전문가)의 특성

노임단가 등급으로 본 Cluster 5(개발전문가)의 최빈치는 고급기술사(37.5%)였다. 최빈치를 기준으로 한 상위급은 31.30%였으며, 하위급은 31.30%였다. Cluster 5(개발전문가)는 4년제 대학 학위 소지자였으며, 4년제 대학 이상인 석사와 박사 졸업생은 9.6%, 하위급인 고등학교 및 전문대학 졸업생은 65.1%였다. 입직 후 진학한 비율은 9.1%였다. 소프트웨어 관련 훈련에는 9.73개, 456.39시간으로 평균보다 많이 참여하였으며, 비관련 훈련에는 평균 11.57개, 255.67시간으로 역시 평균보다 적게 참여하였다.

Cluster 5(개발전문가)가 소지한 자격증의 최빈치는 기사 자격이었으며, 자격을 미소지하고 있는 비율은 11.8%였다. 기사보다 높은 기술사 자격 소지 비율은 2.30%, 산업 기사와 기능사 자격 소지 비율은 44.20%였다. 또한, 프로젝트에 참여한 수는 평균 49.03개였다. Cluster 5(개발전문가)의 총재직년수는 평균 12.78년이었으며, 소프트웨어 개발 직무에 종사한 기간은 11.18년이었다.

〈표 IV-30〉 전문성 발달 수준에 따른 Cluster 별 특성 종합

특성 \ 전문성 등급			Cluster 1(초급개발자)	Cluster 2(중급개발자)	Cluster 3(고급개발자)	Cluster 4(특급개발자)	Cluster 5(개발전문가)
노임 단가 등급	최빈치		초급기술사(45.2%)	중급기술자(39.8%)	중급기술자(37.9%)	중급기술자(32.3%)	고급기술자(37.5%)
	상위급(%)		54.48%	32.70%	28.60%	18.50%	31.30%
	하위급(%)		-	27.6%	33.60%	49.30%	31.30%
학교 교육	최빈치		4년제 대학	4년제 대학	4년제 대학	4년제 대학	4년제 대학
	상위급(%)		13.2%	14.0%	14.6%	12.2%	9.6%
	하위급(%)		78.9%	72.9%	70.8%	71.2%	65.1%
평생 교육	입직후 진학비율		5.3%	8.4%	11.8%	7.0%	9.1%
	SW 관련	훈련개수 (4,24개)	초과	이하	이하	초과	초과
		훈련시간 (375.46시간)	초과	이하	이하	이하	초과

〈계속〉

특성 \ 전문성 등급			Cluster 1(초급개발자)	Cluster 2(중급개발자)	Cluster 3(고급개발자)	Cluster 4(특급개발자)	Cluster 5(개발전문가)
평생 교육	비관련 훈련	훈련개수 (12.74개)	이하	이하	이하	초과	초과
		훈련시간 (375.46시간)	이하	이하	이하	초과	초과
소지 자격증	최고 수준 자격		기사	기사	기사	기사	기사
	자격미소지 비율		46%	31%	25.6%	28.5%	11.8%
	상위급(%)		0.00%	1.90%	3.30%	5.10%	2.30%
	하위급(%)		45.90%	39.40%	36.60%	38.70%	44.20%
프로젝트 수행 참여 개수			7	9.3	11.45	18.03	49.03
직무 경력	총재직년수		5.37	7.40	8.53	10.71	12.78
	SW 개발 년수		4.69	6.65	7.13	9.07	11.18

주) 상위급 : 최빈치 기준 상위 등급 / 하위급 : 최빈치 기준 하위 등급

4. 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계

가. 교육훈련, 자격 및 직무경력의 상관관계 및 다중공선성 진단

전문성 발달 수준과 교육훈련, 자격 및 직무경력에 대해 연구에서 정의한 학교교육, 평생교육, 자격증 소지 자격, 프로젝트 수행 자격, 직무경력 변인들의 상관관계는 .144~.249로 낮은 상관관계를 보였다.

〈표 IV-31〉 교육훈련, 자격 및 직무경력과 전문성 발달 수준의 상관관계

구분	1	2	3	4	5	6
1. 전문성 발달 수준	1					
2. 학교교육	.144**	1				
3. 평생교육	.011	.060	1			
4. 자격증 소지 자격	.164**	.021	.030	1		
5. 프로젝트 수행 자격	.249**	.078	.050	.046	1	
6. 직무경력	.214**	.060	.059	.052	.488**	1

주) *: $p < .05$, **: $p < .01$, ***: $p < .001$

소프트웨어 개발자의 전문성 발달 수준과 교육훈련, 자격증, 직무경험의 관계를 검증하기 위해 다중회귀분석을 실행하기에 앞서 다중공선성을 진단하였다. 이를 위해 허용값(tolerance)와 변량팽창지수(VIF, Variance Inflation Factor)를 사용하였으며, 허용값은 1.0에 근접하는 경우 다중공선성이 없다고 판단하며, 값이 매우 작을 경우 다중공선성을 의심한다. 변량팽창지수의 경우는 허용값에 역수를 취한 값으로, 이 역시 1에 근접하는 경우 다중공선성이 없다고 판단하며, 10 이상일 경우 다중공선성이 있는 것으로 간주한다(최동선, 2003; 김두섭, 강남준, 2000). 상관분석 결과 전문성 발달 수준과 상관관계가 없다고 확인된 평생교육을 제외하고 다중공선성을 확인한 결과, 모두 이상이 없는 것으로 확인되었다. 이에 따라 학교교육, 자격증 소지 자격, 프로젝트 수행 자격, 직무경력 4개 변인과 전문성 발달 수준의 회귀분석을 실시하였다.

〈표 IV-32〉 교육훈련, 자격 및 직무경력과 전문성 발달 수준의 타당성 확보

변량	공차한계(Tolerance)	변량팽창계수(VIF)
학교교육	.993	1.007
자격증 소지 자격	.997	1.003
프로젝트 수행 자격	.763	1.311
직무경력	.765	1.308

주) *: $p < .05$, **: $p < .01$, ***: $p < .001$

나. 전문성 발달 수준과 교육훈련, 자격 및 직무경력에 대한 회귀분석

전문성 발달 수준에 영향을 미치는 교육훈련, 자격 및 직무경력별 분석 결과를 확인해보면, 교육훈련, 자격 및 직무경력에 의한 전문성 발달 수준의 설명력은 10.9%로 나타났다. 개별 변인들을 살펴보면, 프로젝트 수행 자격($\beta = .181$, $p < .001$), 직무경력($\beta = .151$, $p < .05$), 자격증 소지 자격($\beta = .146$, $p < .01$), 학교교육($\beta = .120$, $p < .01$)이 전문성 발달 수준에 유의미한 영향을 갖는 것으로 나타났다.

〈표 IV-33〉 전문성 발달 수준과 교육훈련, 자격 및 직무경력간의 회귀분석 결과

구분	표준화계수	
	β	t
학교교육	.120	2.868**
자격증 소지 자격	.146	3.495**
프로젝트 수행 자격	.181	3.782***
직무경력	.151	3.516*
F	15.669***	
R^2	.109	
R^2_{adj}	.102	

주) *: $p < .05$, **: $p < .01$, ***: $p < .001$

다. 전문성 발달 수준과 노임단가 기준 변인의 상관관계 및 회귀분석

1) 노임단가 기준 변인의 상관관계 및 다중공선성 진단

전문성 발달 수준과 노임단가 기준 변인인 최종학력, 최고 수준 소지 자격증, 종사년수 변인들의 상관관계는 .098~.200으로 낮은 상관관계를 보였다.

〈표 IV-34〉 노임단가 기준 변인과 전문성 발달 수준의 상관관계

구분	1	2	3	4	5
1. 전문성 발달 수준	1				
2. 최종 학력	.117**	1			
3. 최고 수준 소지 자격증	.052	.135**	.072	1	
4. 종사년수	.222**	.060	.091*	.113**	1

주) *: $p < .05$, **: $p < .01$, ***: $p < .001$

소프트웨어 개발자의 전문성 발달 수준과 노임단가 기준 변인인 최종학력, 최고 수준 소지 자격증, 종사년수의 관계를 검증하기 위해 다중회귀분석을 실행하기에 앞서 다중공선성을 진단하였다. 이를 위해 허용값(tolerance)와 변량팽창지수(VIF, Variance Inflation Factor)를 사용하였으며, 허용값은 1.0에 근접하는 경우 다중공선성이 없다고 판단하며, 값이 매우 작을 경우 다중공선성을 의심한다. 변량팽창지수의 경우는 허용값에 역수를 취한 값으로, 이 역시 1에 근접하는 경우 다중공선성이 없다고 판단하며, 10 이상일 경우 다중공선성이 있는 것으로 간주한다(최동선, 2003; 김두섭, 강남준, 2000). 상관분석 결과 전문성 발달 수준과 상관관계가 없다고 확인된 소프트웨어 개발 관련 참여 훈련 프로그램 개수 변인을 제외하고 다중공선성을 확인한 결과, 모두 이상이 없는 것으로 확인되었다. 이에 따라 최종 학력, 최고 수준 소지 자격증, 종사년수 3개 변인과 전문성 발달 수준의 회귀분석을 실시하였다.

〈표 IV-35〉 노임단가 기준 변인과 전문성 발달 수준의 타당성 확보

변량	공차한계(Tolerance)	변량팽창계수(VIF)
최종 학력	.989	1.011
종사년수	.990	1.010

주) *: $p < .05$, **: $p < .01$, ***: $p < .001$

2) 전문성 발달 수준과 노임단가 기준 변인에 대한 회귀분석

전문성 발달 수준에 영향을 미치는 노임단가 기준 변인들의 분석결과를 확인해보면, 노임단가 기준 변인 변인들에 의한 전문성 발달 수준의 설명력은 6.4%로 나타났다. 개별 변인들을 살펴보면, 종사년수($\beta = .211$, $p < .000$), 최종 학력($\beta = .099$, $p < .05$)가 전문성 발달 수준에 유의미한 영향을 갖는 것으로 나타났으며, 소프트웨어 개발 관련 참여 훈련 개수는 전문성 발달 수준에 유의하지 않는 것으로 확인되었다.

〈표 IV-36〉 전문성 발달 수준에 대한 노임단가 기준 변인의 회귀분석

구분	표준화계수	
	β	t
최종 학력	.099	2.305*
종사년수	.211	4.892***
F	11.507	
R^2	.064	
R^2_{adj}	.058	

주) *: $p < .05$, **: $p < .01$, ***: $p < .001$

5. 교육훈련, 자격 및 직무경력 변인의 전문성 등급 결정 크기

소프트웨어 개발자의 전문성 등급을 결정짓는 요인들을 분석하기 위해 판별분석을 실시하였다. 교육훈련, 자격 및 직무경력을 보기 위해 학교교육, 평생교육, 자격증 소지 자격, 프로젝트 수행 자격, 직무경력으로 분석하였으며, 기존 노임단가 기준에서 활용하였던 최종학력, 보유 자격증의 최고 수준, 종사년수로 분석을 실시하였다.

가. 교육훈련, 자격 및 직무경력 변인의 판별모형 적합도 검증

교육훈련, 자격 및 직무경력 판별모형의 적합도 검증 결과는 각 집단 간 혹은 집단내의 공분산 행렬(covariance matrix)의 동일성에 대한 Box' M 결과($F=1.740$, $p=0.000$)가 유의하여 동일성 가정에 위배되지 않았다.

〈표 IV-37〉 교육훈련, 자격 및 직무경력 판별모형 동일성 가정에 대한 Box' M값

구분	Box' M	F
교육훈련, 자격 및 직무경력 판별모형	110.975	1.740***

주) *: $p<0.05$, **: $p<0.01$, ***: $p<0.001$

고유값(eigenvalue)이 클수록 높은 예측력을 가지는데, 0.109의 수준으로 나타나 매우 높은 수치는 아니지만 보통 수준의 전문성 발달 수준을 예측한다고 확인되었다. 정준상관 역시 .314의 수준으로 종속변인 분산의 9.86%(0.314^2)가 독립변인들에 의해 설명력을 가지는 것으로 나타났다. Wilks' Lamda($x^2=65.840$, $p<0.001$)를 통해 통계적으로 유의한 모형임이 증명되었다.

〈표 IV-38〉 교육훈련, 자격 및 직무경력 판별모형의 고유값 및 윌크스람다값

함수	고유값	정준상관	Wilks' Lamda	x^2
1	.109	.314	.879	65.840***
2	.014	.118	.975	12.938
3	.011	.102	.989	5.716
4	.001	.027	.999	.375

주) *: $p < .05$, **: $p < .01$, ***: $p < .001$

나. 교육훈련, 자격 및 직무경력의 상대적 중요도

교육훈련, 자격 및 직무경력 판별모형은 적합성에서 문제없는 것으로 나타나 이를 바탕으로 판별적재값인 구조행렬값으로 모형별 판별력이 높은 변인, 즉 중요도 순서를 확인하였다. 교육훈련, 자격 및 직무경력을 활용한 모형에서 4개의 판별함수모형이 도출되었으며 이 중 통계적으로 유의미한 함수 1모형만을 채택하여 확인한 결과, 전문성 등급을 구분짓는 중요한 변인으로는 프로젝트 수행 자격(.765), 직무경력(.753)가 유의미한 중요 변인으로 확인되었다.

교육훈련, 자격 및 직무경력 판별함수에 대한 학교교육, 평생교육, 자격증 소지 자격, 프로젝트 수행 자격, 직무경력 변인들의 상대적 중요도(판별력)를 종합적으로 표현하는 효력지수는 각 함수별 효력지수의 합으로 표현된다(금상수, 조주현, 2012; 여운승, 2006). 4개의 판별함수를 종합적으로 반영한 효력지수에 의한 변인의 중요도 순서는 프로젝트 수행 자격(0.765)>직무경력(0.753)>학교교육(0.184)>자격증 소지 자격(0.109)>평생교육(0.103)의 순으로 나타났다.

〈표 IV-39〉 교육훈련, 자격 및 직무경력을 활용한 판별함수 결과 요약

구분	1		2		3		4	
	판별계수	판별 적재값	판별계수	판별 적재값	판별계수	판별 적재값	판별계수	판별 적재값
프로젝트 수행 자격	.657	.948*	.294	.162	-1.541	-.198	-.180	-.185
직무경력	.354	.883*	-.193	.087	1.662	.400	-.098	-.205
자격증 소지 자격	-.006	.023	.823	.713*	.155	.156	.451	.527
평생교육	-.067	.094	.389	.308*	-.049	-.073	.239	.180
학교교육	.248	.284	-.580	-.453	-.021	-.062	.790	.843*

주) *: p<.05, **: p<.01, ***: p<.001

주2) 판별계수는 표준화 정준 판별함수 계수를 의미함

다. 교육훈련, 자격 및 직무경력에 의한 전문성 등급 분류 정확률

교육훈련, 자격 및 직무경력을 기준으로 전문성 발달 수준을 분류한 결과, 기존 도구를 통해 도출한 전문성 등급과의 분류정확률(hit ratio)은 43.5%로 나타났다.

〈표 IV-40〉 전문성 등급과 교육훈련, 자격 및 직무경력 모형에 따른 예측집단 분류율

구분		교육훈련, 자격 및 직무경력에 의해 구분된 집단					전체
		1	2	3	4	5	
전문성 등급	Cluster 1 (초급개발자)	11 64.7%	1 5.9%	4 23.5%	0 0.0%	1 5.9%	17 100.0%
	Cluster 2 (중급개발자)	6 7.1%	35 41.7%	34 40.5%	3 3.6%	6 7.1%	84 100.0%
	Cluster 3 (고급개발자)	30 18.2%	18 10.9%	80 48.5%	22 13.3%	15 9.1%	165 100.0%
	Cluster 4 (특급개발자)	18 14.3%	17 13.5%	25 19.8%	41 32.5%	25 19.8%	126 100.0%
	Cluster 5 (개발전문가)	1 2.8%	6 16.7%	8 22.2%	7 19.4%	14 38.9%	36 100.0%

주 1) 사례의 분류 정확률 : 43.5%

주 2) 적어도 하나 이상의 변인에서의 결측치 제외(N=98)

라. 노임단가 기준 변인의 전문성 등급 결정 크기

1) 노임단가 기준 변인 판별모형의 적합도 검증

노임단가 기준 변인 판별모형의 적합도 검증결과, 각 집단 간 혹은 집단내의 공분산 행렬(covariance matrix)의 동일성에 대한 Box' M 결과($F=10.650$, $p<.000$)가 유의하여 동일성 가정에 위배되지 않았다.

〈표 IV-41〉 노임단가 기준 변인 판별모형 동일성 검증에 대한 Box' M값

구분	Box' M	F
노임단가 기준 변인 판별모형	440.036	10.650***

주) *: $p<.05$, **: $p<.01$, ***: $p<.001$

고유값(eigenvalue)이 클수록 높은 예측력을 가지는데, 0.069의 수준으로 나타나 매우 높은 수치는 않지만 보통 수준의 전문성 발달 수준을 예측한다고 확인되었다. 정준상관 역시 .256로 나타나 종속변인 분산의 6.55%(0.256^2)가 독립변인들에 의해 설명력을 가지는 것으로 나타났다. Wilks' Lamda($x^2=50.513$, $p<.001$)를 통해서도 통계적으로 유의한 모형임이 증명되었다.

〈표 IV-42〉 노임단가 기준 변인 판별모형의 고유값 및 윌크스람다값

함수	고유값	정준상관	Wilks' Lamda	x^2
1	.070	.256	.905	50.513***
2	.024	.153	.969	16.212
3	.08	.189	.992	4.229
4	.001	.018	.160	.160

주) *: $p<.05$, **: $p<.01$, ***: $p<.001$

2) 노임단가 기준 변인들의 상대적 중요도

노임단가 기준 판별모형은 적합성에서 문제없는 것으로 나타나 이를 바탕으로 판별적재값인 구조행렬값으로 모형별 판별력이 높은 변인, 즉 중요도 순서를 확인하였다. 노임단가 기준 판별모형에서 4개의 판별함수모형이 도출되었으며 이 중 통계적으로 유의미한 함수 1모형만을 채택하여 확인한 결과, 전문성 등급을 구분짓는 중요한 변인으로는 종사년수(.880)만이 유의미한 중요 변인으로 확인되었다.

노임단가 기준 판별함수에 대한 최종학력, 보유 자격증의 최고 수준, 종사년수 변인들의 상대적 중요도(판별력)를 종합적으로 표현하는 효력지수는 각 함수별 효력지수의 합으로 표현된다(금상수, 조주현, 2012; 여운승, 2006). 4개의 판별함수를 종합적으로 반영한 효력지수에 의한 변인의 중요도 순서는 최종학력(0.624) > 보유자격증의 최고 수준(0.353) > 종사년수(0.190)의 순으로 나타났다. 이상을 보면 노임단가 기준 변인들에서는 전문성을 예측하기 위해 최종학력 변인이 중요함을 알 수 있었다.

〈표 IV-43〉 노임단가 기준 변인 판별함수 결과 요약

구분	1		2		3		4	
	판별계수	판별 적재값	판별계수	판별 적재값	판별계수	판별 적재값	판별계수	판별 적재값
최종학력	.183	.842*	.166	-.212	-.455	-.368	.863	-.331
보유자격증의최고수준	.359	.418	-.429	-.423	.759	.714*	.343	.369
종사년수	.799	.238	-.247	.226	-.436	-.376	-.344	.866*

주) *: p<.05, **: p<.01, ***: p<.001

주2) 판별계수는 표준화 정준 판별함수 계수를 의미함

3) 노임단가 기준 변인에 의한 전문성 등급 분류 정확률

노임단가 기준 변인 변인들을 기준으로 전문성 발달 수준을 분류한 결과, 기존 도구를 통해 도출한 전문성 등급수준과의 분류정확률(hit ratio)는 34.1%로 나타났다. 이는 기존 등급과 동일한 등급에 속한 비율을 의미하는 수치이다.

〈표 IV-44〉 전문성 등급과 노임단가 기준 변인 모형에 따른 예측집단 분류율

구분		노임단가 기준 변인에 의해 구분된 집단					전체
		1	2	3	4	5	
전문성 등급	Cluster 1 (초급개발자)	11 65%	1 6%	4 24%	0 0%	1 6%	17 100%
	Cluster 2 (중급개발자)	13 15.5%	28 33.3%	34 40.5%	3 3.6%	6 7.1%	84 100.0%
	Cluster 3 (고급개발자)	30 18.2%	18 10.9%	70 42.4%	22 13.3%	25 15.2%	165 100.0%
	Cluster 4 (특급개발자)	18 14.3%	17 13.5%	41 32.5%	25 19.8%	25 19.8%	126 100.0%
	Cluster 5 (개발전문가)	3 8.3%	7 19.4%	8 22.2%	6 16.7%	12 33.3%	36 100.0%

주 1) 사례의 분류 정확률 : 34.1%

주 2) 적어도 하나 이상의 변인에서의 결측치 제외(N=98)

6. 연구 결과에 대한 논의

지금까지 제시된 소프트웨어 개발자의 전문성 발달 수준 및 교육훈련, 자격, 직무경력에 관한 연구 결과를 토대로 주요 논의사항을 제시하면 다음과 같다.

가. 소프트웨어 개발자의 전문성 발달 수준

이 연구에서 전문성은 특정 분야에서 최고 수준의 전문가가 지닌 능력으로 정의되며, 전문성은 발달 과정을 거쳐 최고 수준에 도달한다. 따라서 전문성 발달 수준이란 전문성 발달 과정 속에서의 개인의 수준이라고 볼 수 있다. 이 연구에서는 소프트웨어 개발자의 전문성 발달 수준을 측정할 수 있는 기준으로 분석 및 설계, 구현, 테스트, 프로젝트 관리를 구성하였다. 그 결과 소프트웨어 개발자의 전문성 발달 수준은 3.00으로 평균 수준이었으며, 세부적으로는 분석 및 설계(.89), 프로젝트 관리(.74), 테스트(.73) 및 구현(.64)의 순으로 상대적 중요도가 있는 것으로 확인되었다.

526명의 소프트웨어 개발자를 대상으로 42개의 문항, 4개 항목으로 구성된 전문성 발달 수준 측정도구의 타당도를 분석한 결과 측정모형은 타당한 것으로 나타났다. 이 연구의 결과는 소프트웨어 개발자의 전문성을 측정하기 위해 관련 지식, 능력을 소프트웨어의 라이프 사이클인 요구분석, 설계, 구현, 테스트의 영역으로 구분하는 선행연구의 결과를 지지한다(Sonnentag, 2006; Chilton & Hardgrave, 2004; Prasad 외, 2010). 지금까지의 전문성 발달 수준 도구는 객관적 측정 가능한 영역 외 전문성 발달에 영향을 미치는 교육훈련, 경험을 전문성 측정 도구에 직접 포함하거나(Sonnentag, 1996), 창의력(Sonnentag, 1996; Wynekoop 외 (2000), 리더십(Chilton & Hardgrave, 2004), 책임감(Prasad 외, 2010), 유연성(Sonnentag, 1996) 등의 태도적인 변인들을 포함하는 등 여러 변인이 혼재되어 있었다. 이 연구의 결과를 통해 소프트웨어 개발자의 전문성 발달 수준의 경우 지식과 기술이 포함된 4가지 요인으로 전문성 발달 수준을 타당하게 측정할 수 있다는 것으로 확인되었다.

또한, 소프트웨어 개발자의 전문성 발달에 있어 분석 및 설계, 구현, 테스트와 프로젝트 관리가 같은 가중치를 갖지 않는다는 선행연구 결과(Sonnentag, 1993)를 토대로 주성분분석을 실시하여 하위요소들의 가중치를 분석하였다. 그 결과 분석 및 설계(.303), 프로젝트 관리(.250), 구현(.224), 테스트(.223)의 순으로 선행연구 결과를 지지하는 결과가 나타났다. 따라서 소프트웨어 개발자의 전문성은 소프트웨어 개발 생명주기(life cycle)의 영역들을 수행할 수 있는 능력으로 구성되기는 하나, 전문가일수록 분석 및 설계, 프로젝트 관리 능력이 뛰어난 사람임을 의미하였다.

나. 전문성 발달 수준에 따른 등급화

이 연구는 전문성 발달 수준을 확인하는 것에 그치지 않고, Dreyfus & Dreyfus(1980)가 제시한 5단계의 전문성 등급(stage)으로 구분되는지 분석하였다. 연구결과 소프트웨어 개발자의 전문성 발달 수준도 통계적으로 타당하게 5개의 군집으로 구분되었다. 하지만 5개 군집별 평균을 보면 1집단은 1.60, 2집단은 2.38, 3집단은 2.97, 4집단은 3.47, 5집단은 4.13의 특성이 나타났다. Dreyfus & Dreyfus(1980)의 전문성 발달 단계별 특성을 토대로 설문문항을 구성하였기 때문에 군집분석을 통해 분류된 5집단은 5에 가까운 평균이 나와야함에도 4.13에 실제 전문가(expert) 단계라기

보다는 숙련자(proficient)이라고 보아야 할 것이다. 왜 이러한 결과가 나타난 것일까?

이러한 현상은 이 연구에서의 결과만으로 볼 수는 없으며, 이미 전문성 발달에 대한 선행연구에서도 논의되고 있는 부분이다. Fook 외(2000)은 Dreyfus & Dreyfus(1980)의 전문성 발달 단계를 바탕으로 연구하면서 흥미로운 사실을 발견했었다. 일정 기간 해당 분야에 종사하여 경험하면 일정 수준까지 도달할 수 있는 있으나 최고 수준인 expert 수준까지 도달하는데에는 한계가 있다는 것이었다. 또한 대부분의 사람들이 특정 분야에 수십 년 종사함에도 불구하고 expert에 도달하지 못하는 것은 Dreyfus & Dreyfus(1980)가 말한 5단계 외 다른 숙련자 수준(experienced group)과 전문가 수준(expert group)이 상이한 단계가 있을 것이라고 주장했다. 따라서 Fook, Ryan & Hwakins(2000)는 직업전문성 이론(theory of professional expertise)을 제시하면서 Dreyfus & Dreyfus(1977) 모형을 기반으로 하되 초보자 전 단계로서 Pre-study단계, Dreyfus의 마지막 단계인 전문가 단계에서 경험이 많은 숙련자(experienced)와 전문가(expert practice)를 구분하여 총 7단계로 전문성을 보았다. Fook 외(2000)의 연구는 교육훈련, 연습, 경력기간 등으로 도달할 수 있는 것은 experience practice 수준이며 그 이상의 expert 수준으로 가기 위해서는 다른 요소들이 필요하다고 보았다.

따라서 이 연구의 결과가 Dreyfus & Dreyfus(1980)의 전문가(expert)집단과 상이한 부분은 Fook 외(2000)의 직업전문성 이론(theory of professional expertise)으로 설명될 수 있을 것이다.

다. 전문성 등급별 명칭과 특성에 대한 논의

노임단가 등급에서 명명한 초급기술자, 중급기술자, 고급기술자, 특급기술자, 기술사와 Dreyfus & Dreyfus(1980)이 제시한 초보자, 고급입문자, 능숙자, 숙련자, 전문가를 바탕으로 군집분석 결과를 바탕으로 전문성 발달 수준의 평균에 따라 명명하면 Cluster 1은 Cluster 1(초급개발자), Cluster 2는 Cluster 2(중급개발자), Cluster 3은 Cluster 3(고급개발자), Cluster 4는 Cluster 4(특급개발자) 그리고 가장 높은 수준인 Cluster 5는 Cluster 5(개발전문가)이다.

Cluster 1(초급개발자)는 입직 직후부터 5년까지로 전문대학이나 4년제 대학 수준의 초기입직자들의 특성이 나타났다. 이들은 대학졸업자격에 관련 기사 자격 소지자의 특성을 보이며 직업사전에서 제시한 입직 자격의 수준과 비슷한 모습을 보였다. Cluster 2(중급개발자)는 7.4년의 경력기간을 가지고 있으나 학력과 자격증의 수준은 Cluster 1(초급개발자)의 수준과 크게 상이하지 않았다. 다만 참여한 프로젝트의 개수에서 9개로 증가하여 현장에서 다양한 경험을 하는 것이 특징이었다. Cluster 3(고급개발자)는 11.5년 정도의 수준으로 학력은 4년제 대학 수준, 기사 자격 소지자 정도의 특성을 보인다. 다만 석사 이상 학력 소지자의 비율이 높고 자격에서도 미자격소지자의 비중이 낮아지는 특성을 보였다. 비관련 훈련에 많이 참여하여 서서히 관리와 관련된 개발 외 능력을 쌓는 것을 알 수 있었다. Cluster 4(특급개발자)는 11년정도 경력을 쌓은 개발자로 비관련 프로젝트에 평생교육에 높은 참여를 보였다. Cluster 5(개발전문가)는 총 재직년수 13년, 소프트웨어 개발 종사년수 11년으로 전문가에 도달하기 위해 10년 이상의 이상이 필요하다는 법칙을 지지한다(Ericsson, 1997). 이 법칙은 분야를 막론하고 통용되어 사용되고 있는 법칙이다. 소프트웨어 개발자를 대상으로 한 Turley & Bieman(1995)의 연구에서도 우수한 성과자의 경우 평균 9년(4~15년)의 경력기간이 나타났으나 보통수준의 성과자의 경우에는 평균 5년(2~7.5년)의 경력기간이 소요된 것으로 조사되었다. 이 연구 결과에서도 10년이면 전문성 Cluster 4(특급개발자)에 도달한 수준이므로 일정 수준의 성과를 내기 위해서는 평균 10년의 시간이 필요하다는 10년의 법칙을 지지한 결과라고 볼 수 있다.

라. 전문성 발달 수준과 교육훈련, 자격 및 직무경력에 관한 논의

학교교육, 평생교육, 자격증 소지 자격, 프로젝트 수행 자격, 직무경력과 전문성 발달 수준의 관계를 분석한 결과, 평생교육 변인은 전문성 발달 수준과 통계적으로 유의미한 상관이 나타나지 않았으며, 그 외 변인은 .144~.249으로 낮은 상관관계가 나타났으나 노임단가 기준 변인들에 비해서는 다소 높은 수준이었다. 이를 바탕으로 평생교육 변인을 제외한 학교교육, 자격증 소지 자격, 프로젝트 수행 자격, 직무경력과 전문성 발달 수준의 영향을 보기 위해 회귀분석을 실시한 결과, 교육훈련, 자격 및 직무경력의 전문성 발달 수준의 설명력은 10.9%였으며, 프로젝트 수행 자격($\beta=.181$, $p<.001$), 직무경력(β

=.151, $p < .05$), 자격증 소지 자격($\beta = .146$, $p < .01$), 학교교육($\beta = .120$, $p < .01$), 이 전문성 발달 수준에 유의미한 영향을 갖는 것으로 나타났다.

이 연구결과를 살펴보면, 노임단가 기준 등급에서는 사용되지 않는 프로젝트 수행 자격 변인이 전문성 발달 수준에 가장 높은 영향을 갖고 있는 것으로 나타났다. 이는 Sonnentag(1995)의 연구결과를 지지하는 결과이다. Sonnentag(1995) 연구는 우수한 성과자와 보통수준의 성과자의 차이를 분석함에 있어 우수한 성과자의 경우, 보통수준의 성과자에 비해 더 많은 프로젝트를 경험했다는 차이가 밝혔다. 우수한 성과자는 평균 7.4개의 프로젝트에 참여했으며, 보통수준의 성과자는 3.5개의 프로젝트에 참여했다. Sonnentag(1995) 연구 결과는 두 집단 간의 경력기간의 차이보다 프로젝트 참여가 더 중요한 요인이라고 제시하였는데, 이 연구 결과도 프로젝트 수행 자격(.181)이 직무경력(.111)보다 더 높은 영향관계임이 밝혀서 Sonnentag(1995)의 연구 결과를 지지하는 결과가 나타났다. 이러한 결과 역시 소프트웨어 분야의 특수성과 연결해보아야 한다. 소프트웨어 분야의 경우 프로젝트는 장기적으로 수행되는 것도 있지만 일정한 기간이 정해져 있어 프로젝트가 바뀔 때마다 새로운 직무를 수행하게 되거나 새로운 주제를 다루게 되어 다양한 경험들을 접하게 된다. 또한, 같은 프로젝트라도 본인의 직급에 따라 부여되는 직무가 다르기 때문에 여러 프로젝트에 참여하는 것은 일정 수준에 도달하였음을 입증하는 하나의 자격과 같은 역할을 할 수 있었다. 또한, 여러 프로젝트의 경험은 다양한 연습의 과정이 되었고 다양한 연습과 시행착오를 거쳐 개인의 전문성 발달에 정적인 영향을 주게 된 것으로 이해될 수 있다.

이 연구 결과에서 직무경력이 프로젝트 수행 자격보다 낮게 나타난 이유는 무엇일까? Sonnentag(2001) 연구에서는 소프트웨어 분석 및 프로그래밍 업무를 수행하는 우수한 성과자와 보통 수준 성과자를 비교하였다. 일정 수준에 도달하기 위해서는 우수한 성과자는 평균 6.6년 걸리고, 보통수준 성과자는 평균 7.8년이 소요되었지만 이러한 차이는 통계적으로 유의미하지 않았다. 결국 소프트웨어 분야의 경우 직무수행기간은 전문성에 유의미하게 영향을 미치지 않았다고 보고된 것이다. 소프트웨어 분야는 다른 산업에 비해 변화의 속도가 빠르고 새로운 지식의 생성속도가 빠르기 때문이 이전에 보유한 지식은 금방 쓸모없어지는 경향이 있기 때문이다(Sonnentag, 2001). 하지만 Symon(1998)는 경험이 많은 개발자의 경우 정치적, 대립적 문제가 많은 프로젝트를 이끌어가는데 있어 문제를 잘 해결한다고 보았다. 따라서 개인의 개발 및 프로그래밍과

같은 개인 성과에는 경력기간이 영향을 미치지 않으나, 팀관리와 같은 팀성과에는 경력기간이 유의미한 영향을 미치는 것으로 보였다. 이 연구에서는 직무경력이 전문성 발달에 통계적으로 유의미한 정적 관계임이 나타났지만 그 영향력이 가장 낮은 이유는 Sonnentag(2001)의 연구결과와 연계된다고 볼 수 있다.

노임단가 등급에서 사용하는 변인인 최종학력, 보유 자격증의 최고 수준 및 종사년수와 전문성 발달 수준의 관계를 분석한 결과, 최고 수준 소지 자격증은 전문성 발달 수준과 통계적으로 유의미한 상관이 나타나지 않았으며, 최종학력(.117), 종사년수(.222)로 낮은 상관관계가 나타났으며, 노임단가 기준 변인 변인들에 의한 전문성 발달 수준의 설명력은 6.4%로 나타났으며, 종사년수($\beta = .211$, $p < .001$), 최종 학력($\beta = .099$, $p < .05$)이 전문성 발달 수준에 유의미한 영향을 갖는 것으로 나타났다.

이 연구결과는 학력이 전문성에 영향을 미친다는(김난영, 조원혁, 2012; 오석영, 2011) 연구결과를 지지하는 결과였으며, 종사년수가 전문성에 영향을 미친다는(Bruno & McQuarrie, 1992; Reuber & Fisher, 1993; Reuber, 1997), ‘10년의 법칙’ (Ericsson, 2007)의 이론을 지지하는 결과로 해석될 수 있다. 다만, 보유 자격증이 전문성에 정적인 영향을 미친다는 선행연구(김난영, 조원혁, 2012, 김현진, 2005)의 연구결과와는 상반된 결과였다. 보유 자격증이 전문성에 정적인 영향 관계가 나타나지 않은 것은 소프트웨어 분야의 특수성에 기인한 것으로 볼 수 있다. 우리나라 소프트웨어 개발에 대한 인력 진입을 살펴보면 프로그래밍 언어를 다룰 수 있어야 한다는 기본 전제로 인해 진입장벽이 상당히 높은 편이다. 다시 말해, 경영학 등의 학문처럼 누구나 진입 후 경험을 통해 능력을 함양시켜 나아가기 어려운 분야라는 의미이다. 이런 이유로 우리나라 직업사전이나 O*Net에서도 소프트웨어 개발자는 관련 분야 전공자로 기본 자격을 설정하고 있다. 소프트웨어 분야에서 자격증, 특히 국가기술자격이 유의미하지 않게 나온 것은 소프트웨어 개발과 명확하게 일치하는 자격증의 부재도 큰 이유가 되지만 이미 정규교육에서 전공을 통해 기본 역량을 보유하고 있어 입직하거나 입직 이후 승진하는 경우에도 자격증이 큰 역할을 하지 못하기 때문으로 이해될 수 있다. 반면, 현재 존재하는 국가기술자격 또는 민간자격 등이 현업의 직무능력을 확인하는 좋은 도구로 현장에서 활용되고 있을 경우, 전문성과 보유 자격증의 영향 관계는 유의미하게 나타날 것이다.

마. 전문성 등급을 구분하는 관련 변인들의 등급 결정 크기에 대한 논의

학교교육, 평생교육, 자격증 소지 자격, 프로젝트 수행 자격 및 직무경력의 상대적 중요도(판별력)를 종합적으로 반영한 변인의 중요도 순서는 프로젝트 수행 자격(0.765) > 직무경력(0.753) > 학교교육(0.184) > 자격증 소지 자격(0.109) > 평생교육(0.103)의 순으로 나타났다. 이 결과를 살펴보면 전문성 등급을 구분짓는데 프로젝트 수행 자격과 직무경력 0.7이상의 높은 값이 나타났다. 이는 이 두 가지 변인으로 전문성 등급을 충분히 예측할 수 있을 정도로 높은 수준이다. 이 연구 결과는 10년의 법칙과 함께 직무경력과 업무성과, 전문성이 정적인 결과라는 Hunter & Hunter(1984)의 연구를 지지하는 결과이다. 하지만 Quinones et al.(1995), Schmitt & Cohen(1989)와 Tesluk & Jacobs(1998)이 하였듯이 단순히 기간만이 중요한 것이 아니라 그 기간동안 어떤 일들을 수행했는지 역시 중요하다는 연구를 지지하는 결과라고 볼 수 있다. 반면 노임대가 기준 변인인 최종학력, 보유 자격증의 최고 수준, 종사년수의 상대적 중요도(판별력)를 종합적으로 반영한 변인의 중요도 순서는 최종학력(0.624) > 보유자격증의 최고 수준(0.353) > 종사년수(0.190)의 순으로 나타났다. 이 결과는 전문성 발달 수준과 최종학력, 보유 자격증의 최고 수준, 종사년수와 회귀분석 결과와 상이하였지만, 현장의 모습을 대변해주는 결과였다. 실제 현장에서는 한 개발자의 임금을 지불하는 기준으로 노임단가 등급을 활용하고 있으며, 실제 등급기준표의 변인들 중 현장에서 많이 활용하는 것도학력과 자격증이 있다. 이 연구의 결과와 현장의 활용도를 연결하면, 이 3가지 척도만으로 개인의 능력을 측정하였을 경우에는 현재 현장에서 하는 방법인 학력과 자격증이 가장 설명해준다는 것을 지지한다.

V. 요약, 결론 및 제언

1. 요약

이 연구의 목적은 소프트웨어 개발자의 전문성 발달수준과 교육훈련, 자격 및 직무경력의 관계를 구명하는데 있다. 이 같은 목적을 위해 다음과 같은 연구목적을 설정하였다. 첫째, 소프트웨어 개발자의 전문성의 발달 수준을 구명한다. 둘째, 소프트웨어 개발자의 전문성 발달 수준에 따라 등급을 구분한 후, 등급별 특성을 구명한다. 셋째, 소프트웨어 개발자의 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계를 구명한다. 넷째, 소프트웨어 개발자의 전문성 등급 집단을 구분짓는 교육훈련, 자격 및 직무경력의 판별력을 구명한다. 이 연구의 모집단인 소프트웨어 개발자는 소프트웨어 개발 생명주기(Life Cycle) 과정인 분석 및 설계, 구현 및 테스트와 전체 프로젝트를 총괄하는 관리 업무를 수행하는 사람이다. 한국소프트웨어산업협회(KOSA)에 등록된 기업 중 2014년 ‘소프트웨어 기술자 역량인정체계 구축을 위한 연구’에서 활용한 250개 기업 리스트를 활용하여 표집 대상을 선정한 후 이 중 유의표집을 통해 20개의 기업을 선정하였다.

자료수집을 위한 소프트웨어 개발자의 전문성 발달 수준 측정도구는 선행연구와 국가 직무능력표준, TOCIT 등에서 사용한 개발생명주기(life cycle)인 분석 및 설계, 구현, 테스트, 프로젝트 관리 영역으로 연구자가 개발하였다. 예비조사를 통한 문항의 내적합치도 및 구인타당도 분석을 거쳐 본 조사에 사용할 문항을 재추출하였으며, 최종적으로 선정된 문항의 확인적 요인분석 결과를 통해 타당도와 신뢰도를 확보하였다.

자료수집은 2015년 11월 2일부터 11월 13일까지 2주간 진행하여 단순 직무를 수행하는 기능직을 제외한 526부가 최종 분석에 활용되었다. 수집된 자료는 각각의 연구목표에 따라 확인적 요인분석, 주성분분석, K-mean 군집분석, F검증, 다중회귀분석, 판별분석을 활용하여 분석하였다. 추리통계에 대한 통계적 유의성은 0.05를 기준으로 판단하였다.

연구문제 1를 수행하기 위해 확인적 요인분석을 활용하여 소프트웨어개발자의 전문성 발달 수준의 모형을 확정하여 도구를 검증하였으며, 주성분 분석을 활용하여 4가지 구인별 가중치를 고려하여 소프트웨어 개발자의 전문성 발달 수준을 도출하였다. 연구문제 2를 수행하기 위해 k-mean 군집분석을 활용하였으며, 각 등급별 특성을 비교하기 위해 빈도분석, 교차분석을 실시하였다. 연구문제 3은 전문성 발달 수준과 교육훈련, 자격

및 직무경력과의 관계를 분석하기 위해 다중회귀분석을 활용하였다. 끝으로 소프트웨어 개발자의 전문성 등급과 관련 변인들의 판별력을 구명하기 위해 판별분석을 활용하였다.

이 연구에서 사용한 소프트웨어 개발자의 전문성 발달 수준 도구는 소프트웨어의 Life cycle인 요구분석, 설계, 구현, 테스트를 기준으로 분석 및 설계, 구현, 테스트, 프로젝트 관리의 4가지 구인으로 구성하였다. 영역별 필요한 능력은 소프트웨어 개발자의 국가직무능력(NCS, National Competency Standard)와 신자격의 내용 등을 반영하였으며, TOPCIT(Test of Practice in IT)의 일부 프로젝트 관리 내용을 바탕으로 연구자가 개발하거나 재구성하여 활용하였다.

도구를 검증하기 위해 HRD 전문가와 소프트웨어 분야 종사자에게 내용타당도를 검토하였고, 소프트웨어 개발자 100명을 대상으로 예비조사를 실시하여 타당도와 신뢰도를 검토하였다. 이 과정에서 신뢰도가 낮은 임베디드 테스트(테스팅 10) 문항이 삭제하였다. 일부 문항의 경우 탐색적 요인분석 결과 유의하지 않은 결과가 나타났으나 연구의 특성상 해당 영역에서 필수적으로 요구되는 능력이라 통계적으로 유의하지 않다는 이유로 해당 능력을 삭제하기에는 어려움이 있어 1개 문항을 제외하곤 모든 문항을 본조사에 활용하였다.

이 연구의 결과를 요약하면 다음과 같다. 첫째, 소프트웨어 개발자의 전문성 발달 수준은 분석 및 설계, 구현, 테스트 및 프로젝트 관리로 구성되었으며 조사도구에 응답한 값은 평균 3.02였다. 전문성 발달 수준이 높아짐에 따라 분석 및 설계, 프로젝트 관리의 중요도가 높아진다는 선행연구 결과를 반영하기 위해 주성분 분석을 실시하여 영역별 가중치를 도출한 후 전문성 발달 수준을 보정하였다. 영역별 가중치는 분석 및 설계(0.303), 구현(0.224), 테스트(0.223) 및 프로젝트 관리(0.250)였으며, 가중치를 반영하여 보정한 소프트웨어 개발자의 전문성 발달 수준 평균은 3.00으로 유사하게 나타났다.

둘째, 소프트웨어 개발자의 전문성 발달 수준으로 군집분석을 실시한 결과 5개 집단으로 구분되었으며, 전문성 발달 수준의 평균값으로 Cluster 1(초급개발자)는 1.60, Cluster 2(중급개발자) 2.38, Cluster 3(고급개발자) 2.97, Cluster 4(특급개발자) 3.47, Cluster 5(개발전문가) 4.13로 명명하였다. 교육훈련, 자격 및 직무경력을 토대로 등급별 특성을 살펴보면 Cluster 1(초급개발자)는 노임단기 초급기술자 수준이었고 전문대학에서 4년제 대학의 중간 수준(2.89)이었다. 입직 후 상급학교 진학은 5.3%, 소프트웨어 관련 훈련에 평균 5.85개, 비관련 훈련에 평균 9.5개 참여하였다. 평균 기사 자격을 소지하고 있으며, 7개 프로젝트에 참여하고 5.37년 재직하는 것으로 확인되었다.

Cluster 2(중급개발자)는 노임단가 초급기술자 수준이었으며 4년제 대학(2.96)수준이었다. 입직 후 상급학교 진학은 8.4%, 소프트웨어 관련 훈련은 평균 2.34개, 비관련 훈련에 3.37개 참여하였다. 평균 기사 자격을 소지하고 있으며, 9.37개 프로젝트에 참여하고 7.40년 재직하는 것으로 확인되었다. Cluster 3(고급개발자)는 노임단가 중급기술자 수준이었으며, 4년제 대학(2.97)수준이었다. 입직 후 상급학교 진학은 11.8%, 소프트웨어 관련 훈련은 평균 3.26개, 비관련 훈련은 평균 11.57개 참여하였다. 평균 기사 자격 수준이었으며, 11.45개 프로젝트에 참여, 8.53년 재직한 것으로 확인되었다. Cluster 4(특급개발자)는 노임단가 고급기술자 수준이었으며 4년제 대학(3.03)수준이었다. 입직 후 상급학교 진학은 7.0%, 소프트웨어 관련 훈련은 5.08개, 비관련 훈련은 13.3개 참여하였다. 평균 기사 자격 수준이었으며, 18.03개 프로젝트 참여, 10.71년 재직한 것으로 확인되었다. Cluster 5(개발전문가)는 노임단가 고급기술자 수준이었으며 4년제 대학에서 석사학위(3.12) 수준이었다. 입직 후 상급학교 진학은 9.1%, 소프트웨어 관련 훈련 9.73개, 비관련 훈련에 44개 참여하였다. 평균 기사 자격 수준이었으며, 49.03개 프로젝트 참여 11.18년 재직한 것으로 확인되었다.

셋째, 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계를 분석한 결과, 프로젝트 수행 자격, 직무경력, 자격증 소지 자격, 학교교육 순으로 상관관계가 높게 나타났으며, 평생교육은 상관이 없는 것으로 나타났다. 전문성 발달 수준과 교육훈련, 자격 및 직무경력 변인 중 프로젝트 수행 자격 .181, 직무경력 .151, 자격증 소지 자격 .146, 학교교육 .120이 통계적으로 유의미하게 전문성 발달 수준에 영향을 미치는 것으로 나타났다. 반면 전문성 발달 수준과 노임단가 기준 변인간의 관계를 분석한 결과는 노임단가 기준 변인 중에서는 종사년수, 최종학력순으로 상관관계가 높게 나타났으며, 최고 수준 소지 자격증은 상관이 없는 것으로 나타났다. 전문성 발달 수준과 노임단가 기준 변인 중 종사년수 .211, 최종학력 .099가 통계적으로 유의미하게 전문성 발달 수준에 영향을 미치는 것으로 나타났다.

넷째, 교육훈련, 자격 및 직무경력을 활용한 모형에서 효력지수에 의한 변인의 중요도 순서는 프로젝트 수행 자격(0.765)>직무경력(0.753)>학교교육(0.184)>자격증 소지 자격(0.109)>평생교육(0.103)의 순으로 나타났다. 또한, 교육훈련, 자격 및 직무경력을 활용한 모형을 활용한 경우 전문성 등급과의 분류 정확률이 43.5%로 기존 노임단가 변인을 활용한 경우의 분류 정확률인 34.1%보다 높게 나타났다. 노임단가 기준 모형에서 효력지수에 의한 변인의 중요도 순서는 최종학력(0.624)>보유자격증의 최고 수준(0.353)>종사년수(0.190)의 순으로 학력 및 자격증보다 종사년수의 중요도가 더 낮은 것으로 나타났다.

2. 결론

연구의 결과 및 논의를 바탕으로 다음과 같은 결론을 제시하였다.

첫째, 소프트웨어 개발자의 전문성 발달 수준은 개발 생명주기(life cycle)를 반영한 분석 및 설계, 구현, 테스트, 프로젝트 관리로 구성된 측정도구로 측정할 수 있으며 전문성 구인별 가중치를 적용해서 정밀하게 측정할 수 있다. 소프트웨어 개발자의 경력경로는 입직 후 구현의 업무를 수행하다가 테스트, 관리와 함께 최종적으로는 요구 분석 및 설계 직무를 수행하는 경력경로를 거친다. 따라서 도구에 의한 측정값을 활용할 수도 있으나 경력개발 단계에 따라 4단계 구인별 비중을 반영하여 전문성 발달 수준을 측정하는 것도 중요하다.

둘째, 소프트웨어 개발자의 전문성 발달 수준에 따른 등급은 초급개발자, 중급개발자, 고급개발자, 특급개발자, 개발전문가 5단계로 설정할 수 있었고, 초급개발자에서 개발전문가로 갈수록 전문성 발달 수준이 높아졌다. 소프트웨어 개발자의 전문성 발달 역시 일정한 단계를 거쳐 발달된다는 사실이 확인되었다. 전문성 등급이 높아질수록 노임단가 등급은 초급기술자에서 고급기술자로 높아졌으나, 학력은 대체적으로 학사 수준이었다. 다만 전문성 등급이 높아질수록 고등학교와 전문대학 졸업 비율은 낮아졌으며 석사 및 박사의 비율이 높아지는 특성이 나타났다. 입직 후 진학 비율은 전문성 등급에 따른 방향성이 있지 않았으나 소프트웨어 개발 관련 훈련과 비관련 훈련은 전문성 등급이 높아질수록 더 많아지는 경향이 나타났다. 최고 수준의 자격은 기사자격이었으며, 자격 미소지자의 비율은 전문성 등급이 높아질수록 낮아지는 경향성이 나타났다. 프로젝트 수행 참여 개수와 재직년수는 전문성 등급이 높아질수록 높아졌으며, 특히 소프트웨어 개발년수는 개발전문가의 경우 11.18년으로 나타나 10년간 종사하면 해당 분야의 전문가가 될 수 있다는 10년의 법칙과 같은 결과가 나타났다.

셋째, 소프트웨어 개발자의 전문성 발달 수준을 예측하기 위해서는 교육훈련, 자격 및 직무경력 변인이 사용될 수 있다. 학교교육, 평생교육, 자격 및 직무경력 변인을 활용할 경우에는 최종학력, 보유자격증의 최고 수준, 종사년수로 단편적으로 측정하는 것이 아니라 학교교육을 측정함에 있어서는 단순히 학력만이 아니라 지금까지 이수한 학교급들의 전공까지 고려해야 하며, 평생교육에서도 입직 후 상급학교의 진학, 소프트웨어 관련과 비관련 훈련을 모두 반영해서 측정해야 한다. 개인이 일정수준에 도달했다는 것도 검증

된 자격증으로도 인정받을 수 있지만 검증되지는 않았으나 현장에서 인정해주는 다른 요소들도 함께 고려해야 한다. 소프트웨어 개발 분야의 경우 참여했던 프로젝트를 비검증된 요인으로 활용할 수 있었다. 따라서 자격을 측정하기 위해 자격증을 활용할 경우에는 최고 수준 소지 자격증과 함께 자격증의 개수, 유형 등도 함께 고려해서 측정해야 하며, 프로젝트 수행을 측정은 프로젝트에서 수행한 일을 예측할 수 있는 직급과 참여 개수를 고려해서 측정해야 한다. 끝으로 직무경력도 총 재직년수만이 아닌 직급별 수준을 고려해야 측정해야 보다 개인의 전문성을 잘 예측할 수 있는 자료로 활용할 수 있을 것이다.

넷째, 전문성 등급을 구분짓는데 가장 중요한 변인으로는 프로젝트 수행 자격, 직무경력, 학교교육, 자격증 소지 자격, 평생교육 순으로 나타났다. 프로젝트 수행 자격과 직무경력은 전문성 발달 수준과 등급 결정요소 모두에 가장 중요한 변인으로 확인되었다. 현재까지 개인의 역량을 평가함에 있어 학력, 자격증 등 평가하기 용이한 변인들을 주로 활용된 것과 반대되는 결과이다. 하였다. 지금까지 우리나라에서 개인의 능력 또는 역량을 평가함에 있어 가장 많이 활용하고 있는 학력 또는 자격변인은 유의미한 변인이긴 하였으나 다른 변인들에 비해 영향력이 낮게 나타났다. 이는 학사 자격이 입직 자격인 소프트웨어 분야의 특수성도 반영되어 있겠지만, 고등학교 졸업자의 70% 이상이 고등교육에 진학하는 우리나라의 현실을 반영한 결과라고 볼 수 있겠다. 고학력사회에서는 누구나 일정 수준의 교육을 받으므로 더 이상 학력 또는 입직 전 교육시간으로는 개인의 능력을 평가할 수 있는 지표로 활용되기 어려움을 시사할 수 있겠다. 반면 개인이 입직 후 겪은 다양한 경험이 전문성을 발달 시키는데 큰 영향을 미친다는 것은 입직 연령이 낮아지고 있는 현 시대 현상에도 시사점을 제시할 수 있다. 소프트웨어 개발 분야의 경우에는 준비를 위해 교육훈련에 참여하거나 관련 자격증을 취득하면서 입직을 준비하는 것도 중요하지만 우선 입직 한 후 다양한 경험을 쌓아 능력을 함양해 나아가는 것도 바람직한 방법이 될 것이다.

3. 제언

이 연구의 결과에 대한 논의를 토대로 추후 연구를 위한 제언을 제시하였다.

첫째, 소프트웨어 개발자의 전문성 수준을 면밀하게 측정하게 측정하기 위해서는 경력단계별 구인가중치를 다르게 적용해야 한다. 다시말해 경력경로와 영역별 가중치를 연

결하기 위해서는 소프트웨어 개발 전체 대상자에게 같은 가중치를 적용하는 것이 아니라 초급개발자와 중급개발자에게는 구현 및 테스트의 비중을 더욱 높게 하고 고급개발자 이상에는 분석 및 설계, 관리의 비중을 더 높게 적용해야 경력수준에 따른 전문성 수준도 함께 반영될 수 있을 것이다.

둘째, 교육훈련, 자격 및 직무경력 변인을 관찰가능한 다양한 요소들을 포함하여 측정하고자 시도하였지만 추가적인 지표들을 도출하여 더 정확하게 측정해야 한다. 특히 자격증의 경우 소프트웨어 개발에 관련된 국가기술자격명은 국가직무능력표준(NCS) 등에 제시되어 있는 자료를 활용하였지만 국가공인민간자격, 해외자격은 활용되고 있는 자료가 없었으며, 국가공인민간자격과 해외자격을 국가기술자격 기사급으로 동등하게 적용한 것에 대한 한계도 있었다. 또한, 프로젝트 수행에 대해서도 일본 ITSS의 경우에는 프로젝트에서 개인이 참여한 역할과 함께 프로젝트 투입 인력, 프로젝트 수주 금액 등을 면밀하게 조사하여 프로젝트 규모별 역할의 비중을 다르게 적용하기도 하였다. 따라서 후속연구에서는 교육훈련, 자격 및 직무경력을 측정함에 있어 관찰 가능한 다양한 요소들을 적용할 필요가 있겠다.

셋째, 후속연구에서는 전문성 발달 수준과 교육훈련, 자격 및 직무경력의 관계를 분석하는 경우에는 개인의 현재 직무나 근무하고 있는 기업 규모도 고려해서 통제해서 분석할 필요가 있다. 직무에 따라 요구되는 전문성 영역도 상이할 수 있으며, 작은 규모의 회사에서는 낮은 직급이라도 전반적인 소프트웨어 개발 생명주기를 모두 다루고 프로젝트 관리까지 수행해야 하기 때문에 전문성에 차이를 보일 수도 있다.

넷째, 전문성 등급과 교육훈련, 자격 및 직무경력의 차이분석을 통해 등급별 교육훈련, 자격 및 직무경력의 유의미한 차이를 분석할 필요도 있다. 예를 들어 학교교육은 입직부터 초급 및 중급개발자의 전문성 등급 집단을 구분하는 변인일 수는 있으나, 중급이상의 등급에서는 유의미한 변인이 아닐 수 있다. 이러한 결과는 현장에서 소프트웨어 개발자를 채용하는 경우 같은 채용기준을 적용하는 것이 아니라 경력수준 또는 요구되는 전문성 등급에 따라 서로 다른 기준을 적용할 수 있는 근거자료로 활용될 수 있을 것이다.

다섯째, 전문성 발달 수준에 영향을 미치는 결정요인으로 교육훈련, 자격 및 직무경력을 선정하였지만, 인지심리학 및 교육학 분야에서는 객관적인 교육훈련, 자격 및 직무경력에 대한 투입요소과 함께 개인이 얼마나 의지를 가지고 충실하게 참여하는지에 대한 의도적인 연습(deliberate practice)의 매개효과를 보는 것도 필요할 것이다.

참고문헌

- 강민지. (2012). 근거이론에 의한 아동상담자의 전문성 발달 연구. 숙명여자대학교 대학원. 박사학위논문.
- 강지아. (2010). 초등교사를 중심으로 본 전문성의 형성과 발달에 관한 연구. 서울대학교 박사학위논문.
- 강혜영.(2015). 리커트척도 응답 형식 검사의 조건에 따른 기준점설정 방법 비교 연구. 서울여자대학교 일반대학원. 박사학위논문.
- 고경미, 이선경, 심성경.(2014).유사교사의 교수몰입도와 행복감 및 전문성 발달수준과의 관계. 한국교원교육연구, 31(1), 69-94.
- 국가법령정보센터. (2014. 1. 1). 소프트웨어산업진흥법 시행령.
- 국립국어원. (2015). 표준국어대사전.
- 김미란, 김민경.(2008).제조업 생산직 근로자의 숙련향상이 승진과 임금에 미치는 영향.한국직업능력개발원.
- 김재순, 강예지, 장원섭.(2013). 고숙련 소프트웨어 개발자의 일터에서의 성장과 학습에 대한 질적 사례 연구. 직업교육연구, 32(5), 131-149.
- 김재순, 강예지, 장원섭.(2013). 고숙련 소프트웨어 개발자의 일터에서의 성장과 학습에 대한 질적 사례 연구. 직업교육연구, 32(5), 131-149.
- 김정아, 오현석. (2008). 전문성 구성요소의 발달에 관한 연구. 직업능력개발연구, 10(3), 111-134.
- 김정아. (2007). 방송사 PD전문성의 발달과정 및 구성요소에 관한 연구. 서울대학교 대학원. 박사학위논문.
- 김진모 외. (2007). 일반 모델 덧씌우기 방식을 적용한 농촌지도공무원 역량모델 개발. 농업교육과 인적자원개발, 39(2),115-138.
- 김진모, 주대진. (2006). 농촌지도직공무원의 전문능력 개발 체계. 농업교육과 인적자원개발, 38(1), 67-92.
- 김현진. (2005). 금형산업 근로자의 숙련형성에 관한 연구. 서울산업대학교 석사학위논문.

- 나승일, 강순희, 장현진. (2012). 국가기술자격 운영 실태 및 현황 분석. 고용노동부.
- 문승태, 김민배. (2007). 농업교사의 전문성 요인 척도 개발. 농업교육과 인적자원개발, 29(2), 23-46.
- 문용린. (2006). 인적자원개발케적의 개념을 활용한 개인의 성장과정 연구. 한국교육개발원.
- 미래창조과학부·정보통신기술진흥센터·한국생산성본부. (2015). TOPCIT.
- 박기성. (2006). 기업의 생산직 숙련. 제1회 인적자본기업패널 학술대회 자료집.
- 박동열. (2011). 고숙련사회에서의 숙련기술인 육성 방안. 한국직업능력개발원.
- 박준기. (2010). 전문성 발달 단계를 반영한 교사자격기준 개발 및 타당성 분석. 단국대학교 대학원. 박사학위논문.
- 박지윤. (2010). 지역 인적자본 특성이 개인 생산성에 미치는 영향 연구. 이화여자대학교 대학원. 박사학위논문.
- 배을규, 동미정, 이호진. (2011). 전문성 연구 문헌의 비판적 고찰. HRD 연구, 13(1), 1-26.
- 성은모, 배형준, 강용관, 오현석. (2012). 전문성 성장잠재력 검사도구 개발 연구. 한국교육학회 학술대회, 2012, 106-109.
- 성태제. (1991). 목표지향 검사를 위한 준거설정 방법: 오답추출능력에 의한 준거설정
- 성태제. (2011). 준거설정. 서울: 학지사.
- 소정희. (2006). 학교교육에서 'Competency'의 의미와 교육과정적 시사. 한국교육학회 추계학술대회 자료집. 127-139.
- 손영수, 고훈, 한호현, 이덕웅, 신용태. (2006). 국내 SW 기술 인력 분류 및 성장 전략에 관한 연구. 한국통신학회논문지, 31(3).
- 손은정, 유성경, 심혜원. (2003). 상담자의 자기성찰과 전문성 발달. 상담학연구, 4(3), 367-380.
- 신원부. (2013). 공무원 전문성의 영향요인에 관한 연구. 연세대학교 대학원. 박사학위논문.
- 신종호 외. (2007). 연기 전문성 발달 과정에 대한 사례 연구_개인의 능력 계발 영향요인 탐색. 교육심리연구, 21(2), 401-421.

- 오석영. (2011). 학력 간 학습 형태가 숙련향상 및 조직성파에 미치는 영향 분석(제조업 생산직 중심으로). 직업능력개발연구, 14(3)
- 오현석. (2004). 전문성 개발에 영향을 미치는 변인 및 개발 전략에 관한 연구. 서울대학교 신인교수 연구정착금 지원 연구논문.
- 오현석. (2007). 전문성 개발과정 및 핵심요인에 관한 연구. 직업능력개발연구, 9(2), 193-216.
- 오현석 외. (2007). 과학인재의 성장 및 전문성 발달과정에서의 영향 요인에 관한 연구. 한국과학교육학회지, 27(9), 907-918.
- 오현석, 김정아. (2007). 전문성 연구의 주요 쟁점과 전망. 기업교육연구, 9(1), 143-168.
- 오현석, 성은모, 배진현, 성문주. (2009). 최고 수준 전문가와 보통 수준 전문가의 특성 비교 분석. 아시아교육연구, 10(4), 105-135.
- 온기찬. (1996). 전문성의 본질과 직관에서의 역할에 관한 연구, 교육학연구, 34(3), 53-85.
- 위재경. (2015, 4월 21일). 스마트카 시대 융합 인재 양성 방향. 전자신문. Retrieved from http://www.etnews.com/20150421000277?koost=hotnews&mc=d_002_00001
- 유현숙. (2002). 국가 수준의 생애능력 표준 설정 및 학습제체 질 관리 연구(Ⅰ). 한국교육개발원 연구보고서. 서울: 한국교육개발원.
- 윤정일, 김민성, 윤순경, 박민정. (2007). 인간 능력으로서의 역량에 대한 고찰_역량의 특성과 차원. 교육학회지, 45(3).
- 윤형한. (2009). 직업상담자의 전문성 발달과정과 영향요인. 홍익대학교 대학원. 박사 학위논문.
- 이경례, 문혁준. (2013). 영아보육교사의 발달지식, 교사효능감, 전문성 지원환경이 전문성 발달 수준에 미치는 영향. 한국아동교육연구, 13(3), 163-182.
- 이상돈 외. (2012). IT 전문 융합인력 실태분석 및 전망. 한국직업능력개발원.
- 이선. (2009). 숙련개발체제와 숙련개발을 위한 사회적 파트너십. 노동정책연구, 9(4), 187-223.
- 이선. (2010). 교육훈련에서의 수렴현상과 숙련개발체제의 지속적인 다양성. 산업노동연구, 16(2), 305-343.
- 이영현. (2001). 숙련형성 체제 유형 연구. 직업능력개발연구, 4(1).

- 이재경 외. (2014). 대학진학에 필요한 중고교 학력수준과 교육환경에 관한 종단연구. *교육학연구*, 52(1), 27-57.
- 이재경. (2002). 역량기반 교육과정 개발 방법론에 대한 고찰. *교육공학연구*, 18(4), p.31.
- 이재실. (2011). 일터경험학습을 통한 명장의 성장과정 연구. 아주대학교 대학원. 박사학위논문.
- 이종학. (2014). 자기효능감이 긍정적 자산, 직무만족, 조직몰입, 조직시민행동에 미치는 영향. 경희대학교 대학원. 박사학위논문.
- 이지현 외. (2009). *교육학의 이해*. 서울: 학지사.
- 이지혜, 문용린. (2011). 도덕성 발달 연구를 위한 역량중심 접근의 개념화. *인간발달연구*, 18(1), 227-254.
- 임세영, 김주영. (2012). 차량기술사의 전문성 발달과정에 관한 질적연구. *직업교육연구*, 31(2), 289-314.
- 임현우, 오석태. (2014). 조리전공 대학생의 성별과 자격증 수 변인이 경력개발역량 및 진로결정수준에 미치는 영향에 관한 연구. *외식경영학회*, 17(3), 199-216.
- 장혜원. (2014). 한국의 인지적 숙련과 교육의 연계 연구. 한국교원대학교 교육정책전문대학원. 박사학위논문.
- 장홍근 외. (2004). 작업조직과 숙련형성_정보통신서비스산업의 사례. 한국직업능력개발원.
- 전호진, 이영주, 이정훈. (2012). 소프트웨어 개발인력의 직무만족이 조직몰입도와 이직의도에 미치는 영향에 관한 연구. *한국전자거래학회지*, 17(4), 221-242.
- 조동성, 문휘창. (2006). *국가경쟁력*. 서울: 한국경제신문사.
- 주인중. (2008). 2008년도 국가직무능력표준 개발 및 자격체제 구축 재료분야 직무체제 개발. 한국직업능력개발원.
- 최희선, 김주영, 조진환. (2012). 산업기술인력의 경력경로에 관한 연구: 소프트웨어 산업을 중심으로. 서울: 산업연구원.
- 통계청. (2014). *한국표준산업분류 실무 적용을 위한 분류 설명서*.
- 통계청. (2007. 7. 2). *한국표준산업분류. 제 6차 개정*.

- 한국소프트웨어산업협회. (2014). 소프트웨어 사업 가이드북.
- 한상만, 이희수. (2014). 경험학습을 통한 명의의 전문성 발달과정. 평생교육학연구, 20(1), 93-127.
- 허은정. (2011). 학습조직이 교사전문성에 미치는 효과 연구. 서울대학교 대학원. 박사학위논문.
- 홍아정. (2010). 학습을 통한 자아의 재구성 성인학습자의 고등교육 경험 이해. Andragogy Today, 12(4), 185-218.
- 홍애령 외. (2011). 무용수의 전문성 발달과정 및 특성에 관한 연구. 23, 175-193.
- 홍우림. (2012). 초등교사를 중심으로 본 교사 전문성 형성과 발달에 관한 연구. 서울대학교 대학원. 박사학위논문.
- 황미영. (2006). 사회복지 학사교육과정에서의 사회복지사 전문성 개발에 대한 중요 사건분석. 한국사회복지교육, 2(2), 125-152.
- 황수경. (2007). 한국의 숙련구조 변화와 핵심기능인력의 탐색. 한국노동연구원.
- Baylin, L. & Lynch, J. T. (1983). Engineering as a life-long career: Its meaning, its satisfactions, its difficulties. Journal of Occupational Behavior.
- Benderly, B. L.(1989). Everyday intuition. Psychology Today. 34-40.
- Benner, P. (2004). Describe and Interpret Skill Acquisition and Clinical Judgment in Nursing Practice and Education. Science, Technology & Society, Vol. 24, No. 3, June, 188-199.
- Bereiter, C., & Scardamalia, M.(1993). Expert knowledge and how it comes about, surpassing ourselves: An inquiry into the nature and implications of expertise. La Salle, IL: Open Court.
- Bereiter, C., & Scardamalia, M.(1993). Expert knowledge and how it comes about, surpassing ourselves: An inquiry into the nature and implications of expertise. La Salle, IL: Open Court.
- Berk, R. A.(1986). A consumer's guide to setting performance standards on criterion-referenced tests. Review of Educational Research, 56, 137-172.

- Block, J. H. (1978). Standard and criteria: a response. *Journal of Educational Measurement*, 15(4), 291–295.
- Bloom, B. S., & Sosniak, L. A. (1985). *Developing talent in young people*. New York, NY: Ballantine.
- Bonner, N. (2007). Understanding the role of knowledge in the practice of expert nephrology nurses in Australia. *Nursing and Health Sciences*, 9, 161–167.
- Carr, D. (1993). Questions of Competence. *British Journal of Educational Studies*, 41(3), 253–272.
- Chalykoff, J., & Kochan, T. A. (1989). Computer-aided monitoring: Its influence on employee job satisfaction and turnover. *Personnel Psychology*, 42, 807–834.
- Chi, M. T. H., Glaser, R., & Farr, M. J. (Ed.). (1988). *The nature of expertise*. Hillsdale, NJ: Erlbaum.
- Chilton, M.A & Hardgrave, B.C. (2004). Assessing information technology personnel toward a behavioural rating scale. *The DATA BASE for Advances in Information System*, 35(3), 88–104.
- Christiaens, G., Abegglen, J. A., & Gardner, A. (2010). Expert Holistic Nurses' Advice to Nursing Students. *Journal of Holistic Nursing*, 28, 201–208.
- Chu, S. K., & Law, N. (2007). Development of information search expertise : Research students' knowledge of source types. *Journal of Librarianship and Science*, 39(1). 27–40.
- Cizek, G. J., Bunch, M., & Koons, H. (2004). Setting performance standards: Contemporary methods. *Educational Measurement: Issues and Practice*, 23(4), 31–50.
- Dewey, J. (1938). *Experience and education*. New York, NY: Collier Books.
- Dodgdon, A. (2009). From novice to expert: An investigation into the professional development of Rehabilitation Workers through a study of practice in technical rehabilitation interventions. *The British Journal of Visual Impairment*, 27(2). 159–172.

- Dokko, G., Wilk, S. L., & Rothbard, N. P. (2009). Unpacking Prior Experience_How Career History Affects Job Performance. *Organization Science*, 20(1), 51–68.
- Dreyfus, H. L., & Dreyfus, S.E. (1977). Uses and abuses of multiattribute and multi-aspect model of decision making. Unpublished manuscript, University of California, Berkeley.
- Dreyfus, S. E., & Dreyfus, H. L. (1980). A five-stage model of the mental activities involved in directed skill acquisition. Unpublished report, University of California, Berkeley.
- Dreyfus, H. L., & Dreyfus, S. E. (1986). *Mind over machine: The power of human intuition and expertise in the era of the computer*. New York: Free Press.
- Dyke, L. S., Fischer, D. E., & Reuber, A. R. (1992). An Inter-industry examination of the impact of owner experience on firm performance. *Journal of Small Business Management*, 72–87.
- EALTA. (2000). Methods for setting cut scores in criterion-referenced Achievement Tests.
- Engestrom, Y. (2001). Expansive learning at work : Toward an activity theoretical reconceptualization. *Journal of Education and Work*, 14(1). 133–146.
- Engestrom, Y. (2004). 'The new generation of expertise' , in Rainbird, H., Fuller, A., & Munro, A. *Workplace learning in context*, London: Routledge.
- Ericsson, K. A., & Smith, J.(1991). Prospects and limits of the empirical study of expertise : An introduction. 1–38. In Ericsson, K. A., & Smith, J. (Ed.). *Toward a general theory of expertise*. New York, NY: Cambridge University Press.
- Ericsson, K. A., Krampe, R. T., & Tesch-Romer, C.(1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100(3). 363–406.

- Ericsson, K. A. (Ed.).(1996). The road to excellence : The acquisition of expert performance in the arts and sciences, sports, and games. Mahwah, NJ: Erlbaum. (2003). How the expert performance approach differs from traditional approaches to expertise in sport. 360–402. In Starkes, J. L. & Ericsson, K. A. (Ed.). Expert performance in sports : Advances in research on sport expertise. Champaign, IL.
- Ericsson, K. A., & Lehmann, A. C.(1996). Expert and exceptional performance : Evidence on maximal adaptations on task constraints. *Annual Review of Psychology*, 47. 273–305.
- Ericsson, K. A. & Lehmann, A. C.(1999). “Expertise” , *Encyclopedia of Creativity*, Vol.1, pp. 695 ~ 707.
- Ericsson, K. A. (1998). The scientific study of expert levels of performance_ general implications for optimal learning and creativity. *High Ability Studies*, 9(1), 75–100.
- Ericsson, K. A., Patel, V., & Kintsch, W.(2000). How experts’ adaptations to representative task demands account for the expertise effect in memory recall : Comment on Vicente and Wang(1998). *Psychological Review*, 107. 578–592.
- Ericsson, K. A., & Ward, P.(2007). Capturing the naturally occurring superior performance of experts in the laboratory. *Current Directions in Psychological Science*, 16(6). 346–350.
- Ericsson, K. A., Charness, N., Feltovich, P. J., & Hoffman, R. R. (Ed.).(2007). The cambridge handbook of expertise and expert performance. New York, NY: Cambridge University Press.
- Evans, D.(1991). What is a good nurse? *Nursing*, 4. 9–10.
- Evetts J, Mieg H. A., & Felt, U. (2007). Professionalization, Scientific Expertise, and Elitism: A Sociological Perspective. In Ericsson, K. A., Charness, N., Feltovich, P. J., & Hoffman, R. R. (Eds.), The cambridge handbook of expertise and expert performance(pp 105–123). New York, NY: Cambridge University Press.
- Feldman, D. H. (Ed.).(1955). Learning development in nonuniversal theory. *Human Development*, 38(6). 315–321.

- Feltovich, P. J., Prietula, M. J., & Ericsson, K. A. (2007). Studies of Expertise from Psychological Perspectives. In Ericsson, K. A., Charness, N., Feltovich, P. J., & Hoffman, R. R. (Eds.), *The cambridge handbook of expertise and expert performance*(pp 41–68). New York, NY: Cambridge University Press.
- Fook, J., Ryan, M. J., & Hawkins, L. (2000). *Professional expertise: Practice, theory and education for working in uncertainty*. UK: Whiting & Birch Ltd.
- Ford, J. K, Ouinones, M. Sego, D. J., & Speer–Sorra, J. (1992). Factors affecting the opportunity to perform trained tasks on the job. *PERSONNEL PSYCHOLOGY*, 45, 511–527.
- Gardner, Howard(2000). *Intelligence Reframed: Multiple Intelligences for the 21st Century*, NY: Perseus Books, 문용린 역(2001), 『다중 지능: 인간 지능의 새로운 이해』, 서울: 김영사.
- Germin, M., & Tejada, M. (2012). A Preliminary Exploration on the Measurement of Expertise: An Initial Development of a Psychometric Scale. *HUMAN RESOURCE DEVELOPMENT QUARTERLY*, vol. 23, no. 2, 203–232.
- Glaser, R., & Chi, M. T. H.(1988). Overview. In Chi, M. T. H., Glaser, R. & Farr, M. J.(Ed.). *The nature of expertise*. Hillsdale, NJ: Lawrence Ellbaum Associates.
- Glass, G. (1978). Standards and criteria. *Journal of Educational Measurement*, 15, (4), 237–261.
- Grenier, R. S.(2009). The role of learning in the development of expertise in museum docents. *Adult Education Quarterly*, 59(2). 142–157.
- Hambleton, R. K. (1998). Setting Performance standards on achievement tests: Meeting the requirement of Title I. In L. N. Hansche(Ed.), *Handbook for the development of performance standards*(pp. 87–114). Washington, DC: Council of Chief State School Officers.
- Hambleton, R. K., & Pitoniak, M. J. (2006). Setting performance standards.In R.L.Brennan (Ed.),*Educationalmeasurement*(4th ed., pp. 433–470). Westport, CT: American Council on Education/Praeger.

- Herling, R. W.(1998). "Expertise: the Development of an Operational Definition for Human Resource Development" , in Torraco, R. J.(eds.), Academy of Human.
- Heijden, B. I. J. M.(2000). "The Development and Psychology Evaluation of a Multidimensional Measurement Instrument of Professional Expertise" , High Ability Studies, Vol.11 No.1, pp. 9~39.
- Hoffman, R. R., Shadbolt, N. R., Burton, A. M., & Klein, G. (1995). Eliciting knowledge from experts: A methodological analysis. *Organizational Behavior and Human Decision Processes*, 62, 129–158.
- Hoffman, R. R., & Lintern, G. (2007). Eliciting and Representing the knowledge of experts. In Ericsson, K. A., Charness, N., Feltovich, P. J., & Hoffman, R. R. (Eds.), *The cambridge handbook of expertise and expert performance*(pp 203–222). New York, NY: Cambridge University Press.
- Holyoak, K. J.(1991). "Symbolic Connectionism: Toward a Third-Generation Theories of Expertise" , in Ericsson, K. A. & Smith, J.(eds.), *Toward a General Theory of Expertise*, NY: Cambridge University Press.
- Hunter, J. E., & Hunter, R. F. (1984), Validity and utility of alternative predictors of job performance. *Psychological Bulletin*, 96, 72–98.
- ITSS. (2010). *Skill Standards for IT Professionals*, V.3
- Jacobs, R. L.(1988). Systems theory applied to HRD. 27–60. In D. B. Grados. (Ed.), *Systems theory applied to human resource development*. Alexandria, VA: ASTD.
- Keevy, J. K. & Chakroun. B. (2014). *Level Best*. UNESCO.
- Keith, N., & Ericsson, K. A.(2007). A deliberate practice account of typing proficiency in everyday typists. *Journal of Experimental Psychology: Applied*, 13(3). 135–145.
- Kelly, P., Berry, J., & Battersby, D.(2007). Developing teacher expertise : teachers and students doing mathematics together. *Professional Development in Education*, 33(1). 41–65.

- Kidwell, R. E., & Bennett, N. (1994). Employee reactions to electronic control systems: The role of procedural fairness. *Group and Organization Management*, 19, 203-218.
- Kochevar, L. K.(1994). *Generativity of Expertise*, Unpublished doctoral dissertation. Minneapolis, MN: University of Minnesota.
- Kolb, D. A.(1984). *Experiential learning : Experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice Hall.
- Lave, J., & Wenger, E.(1991). *Situated learning: Legitimate peripheral participation*. 전평국·박성선(공역).(2000). *상황학습*. 서울: 교우사.
- Lyneham, j., Parkinson, C., & Denholm, C.(2008). Explicating Benner' s concept of expert practice: Intuition in emergency nursing. *Journal of Advanced Nursing*, 64(4). 380-387.
- Macduffie, J. P., & Kochan, T. A. (1995). Do U.S Firms invest less in training? *Industrial Relations*, 34(2), 147-168.
- Macmahon, C., Helsen, W. F., Starkes, J. L., & Weston, M(2007). Decision-making skills and deliberate practice in elite association football referees. *Journal of Sports Sciences*, 25(1). 65-78.
- Marsick, V. J., & Watkins, K. E.(1990). *Informal and incidental learning in the workplace*. New York, NY: Routledge.
- Mayer, R. E.(1992). *Thinking, Problem solving, Cognition: 2nd*. New York, NY: Freeman.
- McCall, M. W, Lombardo, M. M., & Morrison, A. M. (1988). *The lessons of experience: How successful executives develop on the job*. Lexington, MA: Lexington Books.
- McClelland, D. C. (1973). Testing for Competence Rather than for "Intelligence" . *American Psychologist*, January, 1-14.
- Mieg, H. A. (2007). Social and Sociological Factors in the Development of Expertise. In Ericsson, K. A., Charness, N., Feltovich, P. J., & Hoffman, R. R. (Eds.), *The cambridge handbook of expertise and expert performance*(pp 743-760). New York, NY: Cambridge University Press.

- Moore, D. R., Cheng, M. I. & Dainty, A. R. (2002). Competence, competency and competencies: performance assessment in organisations. *Work Study*, 51(6), 314–319.
- Nunn, R. (2008). A network model of expertise. *Bulletin of Science, Technology & Society*, 28(5). 414–427.
- Prasad. S. (2010). Performance Assessment of Indian Software Professionals. *Journal of Advances in Management Research*, 7(2), 176–193.
- Quinones, M.A., Ford, J.K., & Teachout, M.S.(1995). The relationship between work experience and job performance: A conceptual and meta-analytic review. *Personnel Psychology*, 48, 887–910.
- Reuber, A.R. & Fischer, E.M.(1994). Entrepreneurs' experience, expertise and the performance of technology-based team. *IEEE Transactions on engineering management*, 41(4), 365–374.
- Schmidt, F. L. (2002). The role of general cognitive ability and job performance: Why there cannot be a debate. *Hum. Perform.* 15(1-2), 187-210.
- Schon, D. A.(1983). *The reflective practitioner : How professionals think in action*. New York, NY: Basic Books.
- SFIA. (2011). SFIA 5 framework reference.
- Shepard, L. (1980). Standard Setting Issues and Methods. *Applied Psychological Measurement*, 4 (4), 447–467.
- Shiva, P. (2010). Performance assessment of Indian software professionals. *Journal of Advances in Management Research* vol. 7, no. 2 (2010), p. 176.
- Simonton, D. K. (1977). Creative productivity, age, and stress: A biographical time-series analysis of 10 classical composers. *Journal of Personalityj and Social Psychology*, 35, 791–804.
- Sireci, S. G., Robin, F., & Patelis, T. (1999). Using Cluster Analysis to Facilitate Standard Setting. *Applied Measurement in Education*, 12(3), 301–325.

- Sonnentag, S. (1995). Excellent Software Professionals: Experience, work activities, and perceptions by peers: *Behaviour & Information Technology*, 14, 289–299.
- Sonnentag, S. (1998). Expertise in professional software design. A process study. *Journal of Applied Psychology*, 83, 703–715.
- Sonnentag, S., Niessen, C., & Volmer, J. (2007). Expertise in Software Design. In Ericsson, K. A., Charness, N., Feltovich, P. J., & Hoffman, R. R. (Eds.), *The cambridge handbook of expertise and expert performance*(pp 373–387). New York, NY: Cambridge University Press.
- Spencer, L. & Spencer, S. (1993). *Competence at work: Models for superior performance*. New York: John Wiley & Sons, Inc.
- Swanson, R. A. & Holton III, E. F. (2001). *The Nature of Human Expertise. Foundations of Human Resource Development*, CA: Berrett-Koehler Publishers.
- Tesluk, P. E., & Jacobs, R. R. (1998). Toward an integrated model of work expertise. *Personnel Psychology*, 51, 321–355.
- Van der Heijen, BIJM.(2002).Individual career initiatives and their influence upon professional expertise development throughout the career. *International Journal of Training and Development*, 6(2), 54–79.
- Winch, C. (2010). *Dimensions of expertise: A conceptual exploration of vocational education*. London: Continuum International Publishing Group.
- Wynekoop, J.L. & Walz, D. B.(2000). Investigating traits of top performing software developers. *Information Technology & People*, Vol. 13 Iss 3 pp. 186 – 195.

[부록 1] 설문도구

소프트웨어 개발자의 전문성 발달 수준과 관련 변인에 관한 설문지

소프트웨어 개발자님,

안녕하십니까?

바쁘신 와중에도 소중한 시간을 내어 주셔서 깊은 감사의 말씀 드립니다.

저는 입직 이후 소프트웨어 개발자의 전문성 발달에 영향을 미치는 교육훈련, 자격 및 직무경력에 대한 요인들을 연구하고 있습니다.

설문지는 총 7면이며, 응답하는데 걸리는 소요시간은 약 10분입니다.

조사결과는 통계법 제 13조 2항에 의거하여 익명으로 처리되므로 특정한 개인이나 기업의 특성은 노출되지 않으며, 오직 연구를 위한 자료로만 활용될 것을 약속드립니다.

또한 응답하지 않은 문항이 하나라도 있으면 그 설문지는 분석할 수 없게 되니, 한 문항이라도 빠짐없이 응답하여 주시기를 부탁드립니다. 귀하의 솔직하고 성의있는 응답은 본 연구를 위해 매우 귀중한 자료가 될 것입니다.

응답과 관련하여 문의사항이 있으시면, 아래 연락처로 연락해 주시기 바랍니다. 끝으로 늘 건강과 행복이 가득하시길 기원하며, 바쁘신 데도 불구하고 시간을 내어 설문에 참여해 주심에 진심으로 감사를 드립니다.

2015년

서울대학교 대학원 농산업교육과

지도교수 : 나 승 일

박사과정 : 꺾 미 선

▶ 연구자 연락처: 02-880-4839/ 010-8829-6588 / mskwak11@snu.ac.kr

※ 소프트웨어 개발자 : 소프트웨어 관련 프로젝트 관리, 업무분석 및 설계, 소프트웨어 구현, DB 구축 업무를 수행하는 자

I. 전문성

다음은 소프트웨어 개발자가 갖추어야 하는 능력입니다. 이러한 능력은 소프트웨어 제품의 생명주기(Life Cycle)인 **분석-설계-구현-테스팅**의 단계별 능력과 전반적으로 프로젝트를 관리할 수 있는 **프로젝트 관리** 능력으로 구분될 수 있습니다. <능력수준> 보기를 읽고 본인의 현재 수준과 일치하는 것에 체크 해 주십시오.

<능력수준> 보기

- **매우 잘 못함(1수준)** : 지식과 기술을 배워 이론적으로 알고는 있지만 다른 동료들에 비해 실제 적용하거나 활용하는 업무를 수행하기에는 **어렵다**고 평가되는 수준
- **잘 못함(2수준)** : 지식과 기술을 적용하거나 활용할 수 있어 **하나의 업무**가 주어지지만 다른 동료들에 비해 그 결과에 대해서는 **완벽하지 않다**고 평가되는 수준
- **보통임(3수준)** : 독립적으로 **하나의 업무**를 맡고 있으며, 다른 동료들에 비해 업무 결과에 대해서도 **신뢰**받고, 상황에 맞는 **의사결정과 판단**이 가능하다고 평가되는 수준
- **잘함(4수준)** : 나만의 노하우로 **직관적인 업무 처리**가 가능하며 다른 동료들에 비해 **우수한 결과**를 보인다고 평가되는 수준
- **매우 잘함(5수준)** : 다른 동료들에 비해 판단과 결정이 **무의식적으로 자동화**되어 일어나며, 관련 지식과 기술의 **통합적 사고**가 가능하며 내가 내린 직관적 판단에 대한 **비판적 성찰**까지 가능하다고 평가되는 수준

1. 소프트웨어 개발자의 능력

[분석 및 설계]

	매우 잘 못 함	잘 못 함	보 통 임	잘 함	매우 잘 함
1. 요구사항 확인능력 : 현행 시스템 분석, 분석모델 확인 등 요구사항 분석기법 활용 등	1	2	3	4	5
2. 어플리케이션 설계 능력 : 정적/동적모델 상세설계, 공통 모듈 설계, 타 시스템 연동 설계 등	1	2	3	4	5
3. 화면구현 능력 : UI 요구사항 확인, 설계, 구현 등	1	2	3	4	5
4. 데이터 베이스 요구사항 분석 능력 : 데이터 요구사항 수집, 분석, 정의, 검증 등	1	2	3	4	5
5. 하드웨어 분석 능력 : 하드웨어 자료 수집, 기능 분석, 시험요구사항 도출, 분석 결과 문서화 등	1	2	3	4	5
6. 개념 데이터 모델링 능력 : 주체영역 정의, 핵심개체 정의, 핵심관계 정의, 개념 E-R 다이어그램 등	1	2	3	4	5
7. 펌웨어 분석 설계 능력 : 운영체제 부트 과정 분석, 펌웨어 요구사항 도출 및 분석 등	1	2	3	4	5
8. 임베디드 어플리케이션 분석 설계 능력 : 기술스펙 검토, 어플리케이션 개발환경 구축, 구현 등	1	2	3	4	5
9. 디바이스 드라이버 분석 설계 능력 : 디바이스드라이버 분석, API 모듈 설계 등	1	2	3	4	5
10. 소프트웨어 아키텍처 설계 능력 : 소프트웨어 아키텍처 설계, 적용 아키텍처 접근법 결정 등	1	2	3	4	5
11. 논리 데이터베이스 설계 능력 : 개체 및 관계 상세화, 데이터베이스 정규화, 논리 데이터모델 검증 등	1	2	3	4	5
12. 데이터 품질관리 능력 : 데이터 품질 정책 수립, 품질 진단, 품질 개선, 품질관리 프로세스 정의 등	1	2	3	4	5
13. 데이터 표준화 능력 : 데이터 표준화 정책수립, 데이터 표준 정의, 표준 관리 등	1	2	3	4	5

[구현]

1. 소프트웨어아키텍처 이행 능력 : 소프트웨어 아키텍처 구현, 구현환경 구성, 단위 테스트 실시 등	1	2	3	4	5
2. SQL 활용 능력 : 기본, 고급, 절차형 SQL 작성	1	2	3	4	5
3. 소프트웨어 공학 활용 능력 : 소프트웨어 개발방법론 활용, CASE 도구 활용, 품질요구사항 확인 등	1	2	3	4	5
4. 어플리케이션 구현 능력 : 개발환경구축, 서버프로그램, 배치 프로그램, 어플리케이션 성능 개선 등	1	2	3	4	5
5. 데이터 입출력 구현 능력 : 논리/물리 데이터저장소 확인, 데이터 조작 프로시저 작성/최적화 등	1	2	3	4	5
6. 제품소프트웨어 패키징 능력 : 제품소프트웨어 패키징, 매뉴얼 작성, 버전관리 등	1	2	3	4	5
7. 통합구현 능력 : 연계 데이터 구성, 연계 매카니즘 구성, 내외부 연계 모듈 구현 등	1	2	3	4	5
8. 펌웨어 구현 능력 : 펌웨어 설계문서 분석, 하드웨어 데이터시트 분석, 부트로더 구현 등	1	2	3	4	5
9. 운영체제 이식 능력 : 커널소스 분석, 커널이미지 구현, 부팅 시험 등	1	2	3	4	5
10. 디바이스드라이버 구현 능력 : 개발환경 구성, 디바이스드라이버 구현, 소스 인스펙션 등	1	2	3	4	5
11. 임베디드 애플리케이션 구현 능력 : 기술스펙 검토, 어플리케이션 개발환경 구축, 어플리케이션 구현 등	1	2	3	4	5
12. 네트워크 프로그래밍 구현 능력 : 개발환경 분석, 기능구현, 프로그램 디버깅, 프로그램 최적화 등	1	2	3	4	5
13. 기술문서 개발 능력 : 요구 문서 검토, 기술문서 작성, 사용자 매뉴얼 작성, 사용자 교육 등	1	2	3	4	5

[테스팅]

1. 테스트 계획 능력 : 환경분석, 전략 수립, 계획 수립, 방법론 선정 등	1	2	3	4	5
2. 테스트 환경 구축 능력 : 테스트 환경, 데이터, 도구, 사전 테스트 실행 등	1	2	3	4	5
3. 개발자 테스트 능력 : 개발자 테스트케이스 설계, 개발자 통합 테스트, 개발자 결함 조치 등	1	2	3	4	5
4. 테스트 분석 능력 : 요구사항, 위험요소, 베이스스 등	1	2	3	4	5
5. 테스트 설계 능력 : 테스트 설계 방법 선정, 케이스 설계 및 작성 등	1	2	3	4	5
6. 테스트 실행 능력 : 단위, 통합, 시스템, 인수테스트 실행 등	1	2	3	4	5
7. 테스트 자동화 능력 : 자동화 도구 정의, 실행, 결과 평가 등	1	2	3	4	5
8. 테스트 관리 능력 : 테스트 진척, 결함, 위험, 산출물 등	1	2	3	4	5
9. 테스트 완료 능력 : 테스트 결과 분석, 결과 평가, 완료 보고 등	1	2	3	4	5
10. 임베디드 시스템 테스트 능력 : 단위 테스트, 시스템 통합 테스트, 버그 수정, 변경 관리 등	1	2	3	4	5
11. UI 테스트 능력 : 사용성 테스트, 테스트 결과 보고 등	1	2	3	4	5

[프로젝트 관리]

1. IT 비즈니스 환경 분석 능력 : IT 비즈니스, 5-Force 분석 모델, 소프트웨어OT 분석 등으로 환경 분석	1	2	3	4	5
2. ITSM (IT Service Management) 시스템 활용 능력	1	2	3	4	5
3. IT 비즈니스 커뮤니케이션 능력	1	2	3	4	5
4. IT 기술 관련 문제해결능력	1	2	3	4	5
5. 프로젝트 관리 능력 : 범위, 일정, 원가, 품질, 인력, 의사소통, 위험, 구매, 이해관계자 등 관리	1	2	3	4	5
6. 프로젝트 성과평가 능력 : 프로젝트 진행률 관리 시스템 활용, PMO 활용하여 성과평가	1	2	3	4	5

II. 교육훈련

다음은 소프트웨어 분야에 처음 들어왔던 입직 시점과 현재 시점에서의 귀하의 고등학교, 대학교, 대학원 교육에 대한 것입니다. 다음을 읽고 귀하에게 해당되는 것에 체크해주시시오.

- 1) 입직시점 : 소프트웨어 개발 회사에 입사한 시점
 2) 입직 이후 상급학교에 진학한 경우에만, 현재시점에 응답해주시시오.

구분		입직시점 ¹⁾	현재시점 ²⁾
고등학교	졸업유무	① 졸업 ② 중퇴(년)	① 졸업 ② 중퇴(년)
	유형	① 인문계고 ② 특성화고	① 인문계고 ② 특성화고
	학과	① 소프트웨어 관련 학과 ② 소프트웨어 비관련 학과	① 소프트웨어 관련 학과 ② 소프트웨어 비관련 학과
대학교	졸업유무	① 전문대(2년제) 졸업 ② 전문대(3년제) 졸업 ③ 대졸(4년제) 졸업 ④ 전문대(2년제) 중퇴 (학기 재학) ⑤ 전문대(3년제) 중퇴 (학기 재학) ⑥ 대학(4년제) 중퇴 (학기 재학) ⑦ 해당사항 없음	① 전문대(2년제) 졸업 ② 전문대(3년제) 졸업 ③ 대졸(4년제) 졸업 ④ 전문대(2년제) 중퇴 (학기 재학) ⑤ 전문대(3년제) 중퇴 (학기 재학) ⑥ 대학(4년제) 중퇴 (학기 재학) ⑦ 해당사항 없음
	학과	① 소프트웨어 관련 학과 ② 소프트웨어 비관련 학과 (소프트웨어 비관련 학과의 경우), ① 소프트웨어 관련 교과를 수강한 적 없음 ② 소프트웨어 관련 교과를 수강한 적 있음 (이수한 과목 총학점수)	① 소프트웨어 관련 학과 ② 소프트웨어 비관련 학과 (소프트웨어 비관련 학과의 경우), ① 소프트웨어 관련 교과를 수강한 적 없음 ② 소프트웨어 관련 교과를 수강한 적 있음 (이수한 과목 총학점수)
대학원	졸업	① 석사 졸업 ② 석사 수료 ③ 석사 재학 (학기 재학) ④ 박사 졸업 ⑤ 박사 수료 ⑥ 박사 재학 (학기 재학) ⑦ 해당사항 없음	① 석사 졸업 ② 석사 수료 ③ 석사 재학 (학기 재학) ④ 박사 졸업 ⑤ 박사 수료 ⑥ 박사 재학 (학기 재학) ⑦ 해당사항 없음
	(석사) 전공	① 소프트웨어 관련 학과 ② 소프트웨어 비관련 학과	① 소프트웨어 관련 학과 ② 소프트웨어 비관련 학과
	(박사) 전공	① 소프트웨어 관련 학과 ② 소프트웨어 비관련 학과	① 소프트웨어 관련 학과 ② 소프트웨어 비관련 학과

※ 소프트웨어 관련 학과 : 전산/컴퓨터공학, 응용소프트웨어공학, 정보/통신공학

구분	소프트웨어 개발 관련 있는 연수 및 훈련		소프트웨어 개발 관련 없는 연수 및 훈련 (리더십, 경제/문화 등)	
	이수 프로그램 수	이수 시간	이수 프로그램 수	이수 시간
입직 후	개	시간	개	시간

III. 자격

다음은 귀하가 소지하고 있는 **소프트웨어 개발 관련 자격증**에 대한 내용입니다. 다음을 읽고 등급별 제시된 자격증 중 귀하가 소지하고 있는 자격증의 개수를 적어주십시오.

등 급	소프트웨어 개발 관련 국가기술자격 종목	옆에 제시되어 있지 않은 자격증명	보유하고 있는 자격증 갯수
기술사	정보관리, 컴퓨터시스템응용, 정보통신, 전자계산기, 전자응용, 전자계산기조직응용,		
기사	정보처리, 전자계산기조직응용, 정보통신(산업), 정보보안, 전자계산기		
산업기사	정보처리, 사무자동화, 정보통신, 정보보안		
기능사	정보기기운용, 정보처리, 전자계산기		
국가공인민간	SIS(정보보호전문가), 데이터이키텍처전문가, 정보시스템감리사, 정보보호전문가, PC활용능력평가시험, 정보시스템감리사, 리눅스마스터, 인터넷정보관리사, 디지털정보활용능력(DIAT), 컴퓨터운용사, PC정비사, PC Master, 정보기술자격시험(ITQ), 정보기술프로젝트관리전문가 (IT-PMP)		
해외	CISA(국제공인정보시스템 감사사), CISSP(국제공인정보시스템 보안전문가), OCP, SCJP, MCDBA, OCP-DBA, CCIE, CSA		

IV. 직무경력

2. 귀하가 지금까지 근무한 경력(근무기간 및 프로젝트 참여)에 대해 기술해주십시오.

직급	근무기간 (년)		프로젝트 참여 수 (개)
	소프트웨어 개발 분야	소프트웨어 개발 분야 외	
1) 사원 (소프트웨어 기술자 등급분류 상 초급기술자에 해당)			
2) 대리 (소프트웨어 기술자 등급분류 상 중급기술자에 해당)			
3) 과장 및 차장 (소프트웨어 기술자 등급분류 상 고급기술자에 해당)			
4) 부장 및 팀장 (소프트웨어 기술자 등급분류 상 특급기술자에 해당)			
5) 임원 (소프트웨어 기술자 등급분류 상 특급 또는 가솔자에 해당)			

※ 소프트웨어 개발 분야 : 프로젝트 관리, 업무분석 및 설계, 소프트웨어 구현, DB 구축 업무

※ 다음은 **일반적인 사항**에 대한 질문입니다. 해당 번호에 체크해주시거나 정확한 숫자를 기입해 주시기 바랍니다.

3. 귀하가 담당하고 있는 소프트웨어 개발 직무는 무엇입니까?

- | | |
|------------|-------------|
| ① 프로젝트 관리 | ② 업무분석 및 설계 |
| ③ 소프트웨어 구현 | ④ DB 구축 |

4. 귀하의 회사는 위의 업무를 분담하여 진행합니까?

- ① 개발인원이 적어 한 팀에서 위의 업무를 모두 수행하는 편이다
- ② 개발인원이 많아 한 팀에서 하나의 세부업무를 맡아 하는 편이다.

5. 귀하는 소프트웨어 기술자의 노임단가 기준 중 어느 등급에 속하십니까?

- | | |
|---------|--|
| ① 기술자 | · 기술사 자격증 보유 |
| ② 특급기술자 | · 고급기술자 자격 취득 후 3년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람 |
| ③ 고급기술자 | · 중급기술자 자격 취득 후 3년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람
· 박사학위를 가진 자로서 기사자격을 취득한 자 |
| ④ 중급기술자 | · 기사 자격을 취득한 자로서 3년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람
· 산업기사 자격을 취득한 자로서 7년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람
· 기사자격을 취득한 자로서 석사학위 취득 후 2년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람
· 기사 자격을 취득한 자
· 산업기사 이상의 자격을 취득한 자 |
| ⑤ 초급기술자 | · 전문학사 이상의 학위를 가진 자
· 고등학교 졸업한 후 3년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람
· 산업기사의 자격을 취득한 자로서 4년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람 |
| ⑥ 고급기능사 | · 기능사의 자격을 취득한 자로서 7년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람
· 산업기사의 자격을 취득한 자 |
| ⑦ 중급기능사 | · 기능사의 자격을 취득한 자로서 3년 이상 소프트웨어 기술 분야에서 일정기간 경력을 갖추거나 근무한 사람 |
| ⑧ 초급기능사 | · 기능사의 자격을 취득한 자 |

바쁘신 와중에도 설문에 응답해주셔서 진심으로 감사드립니다.

빠지신 문항이 없는지 다시 한 번 확인 부탁드립니다.

[부록 2] 측정도구의 예비조사 분석 결과

<부록 2-1> 소프트웨어 개발자의 전문성 신뢰도 분석 결과(N=100, 신뢰도 $\alpha = .983$)

구분	문항번호	수정된 항목-전체 척도 간 상관계수 (항목별)	문항제거 시 신뢰도 계수 (항목별)	문항제거 시 신뢰도 계수 (전체문항)
분석 및 설계 ($\alpha = .963$)	1	.826	.959	.982
	2	.805	.960	.982
	3	.611	.964	.982
	4	.809	.960	.982
	5	.808	.960	.982
	6	.836	.959	.982
	7	.752	.961	.982
	8	.746	.962	.982
	9	.790	.960	.983
	10	.809	.960	.982
	11	.878	.958	.982
	12	.856	.959	.982
	13	.877	.958	.982
구현 ($\alpha = .952$)	1	.718	.949	.982
	2	.680	.950	.982
	3	.741	.948	.982
	4	.718	.949	.982
	5	.836	.946	.982
	6	.753	.948	.982
	7	.823	.946	.982
	8	.786	.947	.982
	9	.756	.948	.982
	10	.828	.946	.982
	11	.790	.947	.982
	12	.759	.948	.982
	13	.632	.951	.982
테스팅 ($\alpha = .946$)	1	.798	.939	.982
	2	.810	.939	.982
	3	.764	.941	.982
	4	.818	.939	.982
	5	.843	.938	.982
	6	.765	.941	.983
	7	.763	.941	.982

구분	문항번호	수정된 항목-전체 적도 간 상관계수 (항목별)	문항제거 시 신뢰도 계수 (항목별)	문항제거 시 신뢰도 계수 (전체문항)
테스팅	8	.820	.939	.982
	9	.786	.940	.982
	10	.585	.948	.983
	11	.637	.945	.983
프로젝트 관리 ($\alpha = .943$)	1	.762	.940	.982
	2	.842	.930	.982
	3	.818	.933	.982
	4	.867	.927	.982
	5	.818	.933	.982
	6	.853	.929	.982

<부록 2-2> 소프트웨어 개발자의 전문성 확인적 요인 분석 결과

구분	비표준화 계수	표준화 계수	표준오차	C.R
분석 및 설계 ← 전문성	1	.971		
구현 ← 전문성	.855	.923	.046	18.676***
테스팅 ← 전문성	.679	.852	.048	14.243***
프로젝트 관리 ← 전문성	.410	.817	.032	12.709***

주 1) $\chi^2=6.424(df=2, p=.040)$, CFI=.988, TLI=.965, RMSEA=.150

주 2) 전체 문항에 대한 분석 결과임

[부록 3] 전문성 등급에 따른 관련 변인의 차이분석 결과

〈부록 3-1〉 노임단가 등급과 전문성 등급의 교차 분석 결과

변인		Cluster 1 (초급개발자)	Cluster 2 (중급개발자)	Cluster 3 (고급개발자)	Cluster 4 (특급개발자)	Cluster 5 (개발전문가)	전체
노임단가 등급	초급기술자	14 45.2%	32 32.7%	46 28.6%	24 18.5%	2 6.3%	118 26.1%
	중급기술자	9 29.0%	39 39.8%	61 37.9%	42 32.3%	8 25.0%	159 35.2%
	고급기술자	5 16.1%	18 18.4%	28 17.4%	34 26.2%	12 37.5%	97 21.5%
	특급기술자	1 3.2%	6 6.1%	13 8.1%	20 15.4%	7 21.9%	47 10.4%
	기술사	2 6.5%	3 3.1%	13 8.1%	10 7.7%	3 9.4%	31 6.9%
계		31 100.0%	98 100.0%	161 100.0%	130 100.0%	32 100.0%	452 100.0%

〈부록 3-2〉 최종학력과 전문성 등급의 교차 분석 결과

변인		Cluster 1 (초급개발자)	Cluster 2 (중급개발자)	Cluster 3 (고급개발자)	Cluster 4 (특급개발자)	Cluster 5 (개발전문가)	전체
최종학력	고졸	2 5.3%	3 2.8%	7 3.8%	2 1.4%	2 4.7%	16 3.1%
	전문대졸 (2, 3년)	3 7.9%	12 11.2%	20 10.8%	15 10.8%	2 4.7%	52 10.2%
	4년제 졸	30 78.9%	78 72.9%	131 70.8%	99 71.2%	28 65.1%	366 71.5%
	석사졸	3 7.9%	14 13.1%	25 13.5%	22 15.8%	11 25.6%	75 14.6%
	박사졸	0 0.0%	0 0.0%	2 1.1%	1 0.7%	0 0.0%	3 0.6%
계		38 100.0%	107 100.0%	185 100.0%	139 100.0%	43 100.0%	512 100.0%

〈부록 3-3〉 최종학력과 전문성 등급의 교차 분석 결과

변인		Cluster 1 (초급개발자)	Cluster 2 (중급개발자)	Cluster 3 (고급개발자)	Cluster 4 (특급개발자)	Cluster 5 (개발전문가)	전체
입직 후 상급학교 진학	비진학	35 92.1%	97 90.7%	161 86.6%	130 90.9%	37 84.1%	460 88.8%
	진학	2 5.3%	9 8.4%	22 11.8%	10 7.0%	4 9.1%	47 9.1%
계		38 100.0%	107 100.0%	186 100.0%	143 100.0%	44 100.0%	518 100.0%

〈부록 3-4〉 최종학력과 전문성 등급의 교차 분석 결과

변인		Cluster 1 (초급개발자)	Cluster 2 (중급개발자)	Cluster 3 (고급개발자)	Cluster 4 (특급개발자)	Cluster 5 (개발전문가)	전체
평생교육	소프트웨어 관련 훈련 개수	5.85	2.34	3.26	5.08	9.73	4.24
	소프트웨어 관련 훈련 시간	662.72	346.45	395.19	289.10	456.39	375.46
	비관련 훈련 개수	9.48	3.37	11.57	13.30	44.42	12.74
	비관련 훈련 시간	53.00	81.96	255.67	569.03	136.30	289.36

〈부록 3-5〉 최종학력과 전문성 등급의 교차 분석 결과

변인		Cluster 1 (초급개발자)	Cluster 2 (중급개발자)	Cluster 3 (고급개발자)	Cluster 4 (특급개발자)	Cluster 5 (개발전문가)	전체
소지 자격	미소지	17 45.9%	32 30.8%	47 25.7%	39 28.5%	18 41.9%	153 30.4%
	기능사	0 0.0%	4 3.8%	7 3.8%	6 4.4%	0 0.0%	17 3.4%
	산업기사	0 0.0%	5 4.8%	13 7.1%	8 5.8%	1 2.3%	27 5.4%
	기사	20 54.1%	61 58.7%	110 60.1%	77 56.2%	23 53.5%	291 57.7%
	기술사	0 0.0%	2 1.9%	6 3.3%	7 5.1%	1 2.3%	16 3.2%
계		37 100.0%	104 100.0%	183 100.0%	137 100.0%	43 100.0%	504 100.0%

〈부록 3-6〉 전문성 등급별 프로젝트 수행 개수

변인	Cluster 1 (초급개발자)	Cluster 2 (중급개발자)	Cluster 3 (고급개발자)	Cluster 4 (특급개발자)	Cluster 5 (개발전문가)	전체
참여 프로젝트 개수	7.00	9.34	11.45	18.03	49.13	15.20

Abstract

The Relationships among the Development Level of Expertise, Training and Development, Qualification and Work Experience for Software Developers

by Misun Kwak

*Dissertation for the Doctor of Philosophy in Education in the
Graduate School of Seoul National University, Korea, 2016*

Major Advisor : Seung-II Na, Ph. D

The purpose of this study was to identify the relationship among the development level of expertise, training and development, qualification and work experience for software developers. The specific objectives were to identify the level of the development of expertise for software developers, to divide the group of expertise based on the development level of expertise, to identify the relationship among the development level of expertise, training and development, qualification and work experience, and to identify the difference between the discriminant effect of training and development, qualification and work experience measured in general and measured in more detail with new measurement method. The population for this study was software developers who perform task such as analyzing, planning, programming, testing, and managing the whole projects. This study has selected 20 industries from the entire software industry for the sampling that Korea Software Industry Association has.

A survey questionnaire was conducted to measure variables of this study. It consisted of the development level of expertise scale, training and development scale, qualification scale, and work experience scale. In this study, the development level of expertise for software developers was developed including analysis and planning, programming, testing and project management. The scales were validated and had appropriate level of reliability verified by content analysis, exploratory factor analysis, reliability test, and confirmatory factor analysis.

The data were collected through mail and online survey from November 2nd to 13rd, for two weeks. A total of 619 out of 681 questionnaires were returned, of which 526 were used for analysis after data cleaning. Data were analyzed through descriptive statistics, correlation analysis, exploratory factor analysis, confirmatory factor analysis, t-test, multiple regression analysis and discriminant analysis. The whole process of the data analysis was accomplished by using the SPSS 22.0 and AMOS 18.0.

To analyze the relationships among the development level of expertise, training and development, qualification and work experience for software developers, the level of education, the highest level of qualification and the whole year of work experience, which measured in wage were used. School education measured in the level of education and major, lifelong learning measured in high education after entering the job field and training, qualification of certification measured in the level of certification, number and type, qualification of project experience measured in project experience of each position, and work experience measured in the highest position and year of each position were also used.

The findings of the study were as follows: first, the weight of each factors of development level of expertise for software developers were analysis and planning(0.303), programming(0.224), testing(0.223), and project management(0.250). The development level of expertise for software developers was 3.00. Second, the development level of expertise for software developers could be divided by five groups: level 1(1.60), level 2(2.38), level 3(2.97), level 4(3.47) and level 5(4.13). Third, factor loading of the whole year of work experience was $\beta=0.211$, factor loading of the level of education was $\beta=0.099$, and the factor loading of the highest level of qualification was not significant. The factor loading of the qualification of project experience was $\beta=0.181$, The factor loading of the work experience was $\beta=0.151$, the factor loading of the qualification of

certification was $\beta=0.146$, the factor loading of the school education was $\beta=0.120$, and the factor loading of the lifelong learning was not significant. Fourth, with factors for measuring in wage, the level of education is the most significant variable in classification of expertise. However, with training and development, qualification and work experience, the qualification of project experience is the most significant variable in classification of expertise.

Based on the findings of the study, major conclusions of this study were as follows: first, the development level of expertise was related with career path of software developer. Second, the development process of the expertise of software developer developments in stages which are different traits. Third, school education, lifelong learning, qualification of certification, qualification of project experience and work experience were used for predicting the development level of expertise of software developer instead for the level of education, the highest level of qualification and the whole year of work experience which measured in wage.

Some recommendations for future researches were suggested: first, expertise need to be measured with affective characteristic factors like passion, creativity, and will. Second, further research is needed to investigate the relationship between the development of expertise and environment factors like family, work, and field. Third, by considering the finding of this study, further research needs to be distinguished with the development process of expertise in another field, not software.

keywords : Software developers, Expertise, Development of expertise, Training and development, Qualification, Work experience

Student Number : 2011–23466